Communications on Applied Electronics (CAE) – ISSN : 2394-4714
Foundation of Computer Science FCS, New York, USA
International Conference on Communication Technology (ICCT 2015) – www.caeaccess.org

Spatial Domain Video Encryptionand Authentication using Chaotic Maps and Secure Whirlpool Hash

Karan Nair
K.J.Somaiya College of Engg.
Vidyavihar, Mumbai

Janhavi Kulkarni
K.J.Somaiya College of Engg.
Vidyavihar, Mumbai

Karan Asher
K.J.Somaiya College of Engg.
Vidyavihar, Mumbai

Prof. Vicky Chheda
K.J. Somaiya College of Engg.
Vidyavihar,
Mumbai

Dr.Jonathan Joshi
Eduvance
110, Shiv Shakti Industrial Estate, Ghatkopar West,
Mumbai

ABSTRACT
Digital video is one of the most popular multimedia data exchanged over the internet. Previous cryptography studies have focused on text data. The encryption algorithms developed for text data may not be suitable to multimedia applications because of large sizes of video. We present an algorithm in which a video file is encrypted by considering each frame as acolour image. Each video frame is broken down into RGB planes. Chaotic mapping algorithms are applied on all the planes of each frame, the temporal and horizontal domain of the video. For end-to-end authentication, semantically secure whirlpool hash has been used. Data integrity is verified using XOR hash which can detecttampering of data at any intermediate node. The algorithm was run on different videos, and satisfying results were obtained.

General Terms
Encryption, multimedia security, video processing

Keywords
Authentication, chaotic maps, encryption, hash, integrity, multimedia, spatial domain, video, whirlpool, XOR

1. INTRODUCTION
Encryption is the process of encoding messages or information so that the original data, or plaintext, cannot be intercepted by an attacker. With the development of both computer and internet technology, multimedia data is being used widely in applications like video-on-demand, video conferencing, forensics, surveillance, military drone feeds. Textual data has its own unique characteristics. Multimedia data, like videos suffer from spatial and temporal redundancies, which gives an attacker extra clues for deciphering the ciphertext. It is imperative to exploit these redundancies to make the ciphertext look completely random. In real time applications, it is necessary to have fast encryption algorithms which doesn't cause any lag or delay. In this paper, we propose an algorithm which uses chaotic maps for encryption of the video. Chaotic maps have been widely investigated over the last decade to meet the increasing demand of fast encryption. It is a diffusion strategy used for permutation of data.Due to the properties of high initial-value-sensitivity, appearing completely random and having superior performance in terms of speed and complexity [6], chaotic maps are suitable for data encryption. Each frame is considered as an image, it is split up into its RGB planes and divided into 8x8 macroblocks. Chaotic maps are applied on the differentplanes in the video to shuffle the macroblocks. A different set of chaotic maps are applied to achieve maximum diffusion of original data. Encryption, by itself can protect the confidentiality of the message, but other techniques are needed to protect the authenticity and integrity of the data. Hashing schemes have been used for authentication and integrity check. Whirlpool hash is used for authentication of the video at the receiver while integrity is checked using XOR hash. Provision has been made to check integrity of the encrypted video at every intermediate node to immediately find out if any corruption or tampering of data has occurred. The algorithm has been implemented in MATLAB. Performance analysis of individual components of the process as well as figures of original and encrypted videos has been shown.

2. RELATED WORK
A straightforward, but naïve approach for video encryption is to consider it as text data. There are several existing algorithms based on AES/DES/IDEA for secure transmission of video. These are the most secure algorithms, but very slow owing to the large amount of data a video contains. The paper [1]proposes a modified AES algorithm which achieves increased performancebut it doesn't account for correlation between pixels in the temporal domain. Many algorithms have been proposed where encryption of the video is done in the frequency domain. Discrete cosine transform (DCT) is used to convert the video from spatial to frequency domain. In a particular DCT block, most of the energy is concentrated in the DC coefficientsand very few AC coefficients. Thus frequency selective algorithms are used for encryption.The algorithm proposed by C. Narsimha et al[2] performed encryption of the first ten coefficients followed by permutation of DC and AC coefficients. The paper by Changgui Shi et al[3] proposed an encryption algorithm named video encryption algorithm (VEA) which uses simple XOR of the sign bits of the DCT coefficients using a secret key. The main disadvantage of domain conversion is the extensive amount of time needed for conversion. It also fails to account for the complexity involved in the transform and quantization losses in the frequency domain due to floating point arithmetic. They also don't use standard cryptographic algorithms, and hence their security is very low[4]. Authentication and integrity form two secondary goals of an encryption algorithm. The algorithms mentioned above don't

have any infrastructure for verification of the source or the video content.

3. PROPOSED ALGORITHM

3.1 Chaotic Maps

The idea of using chaos for encryption is certainly not new and can be traced to the Claude Shannon's paper in 1952. It suggests mixing of data and transformations which depends on their arguments in a sensitive manner[5].Chaos theory is a scientific discipline that focuses on the study of nonlinear systems that are highly sensitive to initial conditions that is similar to random behaviour and continuous system. They are deterministic yet they can be unpredictable in nature. This highly unpredictable and pseudo-random nature of chaotic output is the most attractive feature that leads to many novel applications[6].

3.1.1 Rectangular Chaotic Maps

The algorithm considers each frame as a colour image. They are nothing but two dimensional matrices of some height and width. Conventional chaotic algorithms like Baker's chaotic map and Arnold's cat map work on square matrices [5][10]. They can be used on a video frame provided the video frame is made square. Since video frames are inherently rectangular in size (aspect ratio ≠ 1:1), we need a rectangular chaotic map for encryption of data. A rectangular chaotic transform has been proposed in [7]which is an extension of the Arnold's cat map. It is given by equation (1). Let $gcd(m,n)$ is the greatest common divisor of *m* and *n.*

$$F: \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} mod \begin{bmatrix} H \\ W \end{bmatrix} \quad (1)$$

$$st. \begin{cases} gcd(a, p_h) = 1 \\ b \ (mod\ p_h) = 0 \ or \ c(mod\ p_w) = 0 \\ gcd(d, p_w) = 1 \\ gcd\big((ad - bc), p\big) = 1 \end{cases}$$

where H and W are the heightand width of the video frame respectively; (x, y) is the original pixel position, (x', y') is the mapped position of (x, y); $p = gcd(H, W)$, $p_h = H/p$and $p_w = W/p$. The matrix $A = [(a, c)^T, (b, d)^T]$ is called the transformation matrix of the chaotic transform.The properties and algorithm for effectively generating the coefficient matrix is given below [7].

*Property*Suppose A_0 is a coefficient matrix of 2D rectangular transform. For any positive integers e and f satisfying $gcd(e, H) = 1$and $gcd(f, W) = 1$, let

$$A_1 = \begin{bmatrix} e & 0 \\ 0 & f \end{bmatrix} \times A_0, A_2 = A_0 \times \begin{bmatrix} e & 0 \\ 0 & f \end{bmatrix} \quad (2)$$

Then A_1andA_2 are coefficient matrices of the 2D rectangular transform. Based on this property a coefficient matrix can be created using random integers which depends on the height and width of the frame. The random integers in the algorithm are generated using a pseudo random number generator (PRNG) which is seeded with a value obtained from a password entered by the user.

3.1.1.1 Generation of Transformation Matrix

(1) Generate 8 random integers$r_i (1 \le i \le 8)$ using a seed value. The last random number is i.e. r_8 is given as output as a new seed value. Set $p = gcd(H, W)$and$p_w = W/p$. Generate two integer sequences $\{h_0, h_1 \dots, h_{l-1}\}$ and $\{w_0, w_1 \dots, w_{l-1}\}$ such that $gcd(h_i, H) = 1$and$gcd(w_i, W) = 1, (1 \le i \le l)$. $l$is set to 40 in this algorithm.

(2) Let$b_0 = r_5$, $c_0 = p_w \times r_6$ and $j = r_7 (mod\ l)$. Construct a special matrix A_0 as,

$$A_0 = \begin{bmatrix} 1 & b_0 \\ c_0 & b_0 c_0 + w_j \end{bmatrix}$$

(3) Let $j_i = r_i (mod\ l)$, $(1 \le i \le 4)$. Then calculate the final coefficient matrix using property described in (2) as follows,

$$A = \begin{bmatrix} h_{j1} & 0 \\ 0 & w_{j2} \end{bmatrix} \times A_0 \times \begin{bmatrix} h_{j3} & 0 \\ 0 & w_{j4} \end{bmatrix}$$

3.2 Implementation of Chaotic maps

The main aim of using chaotic maps is to break the correlation amongst pixels in the video. In an image, chaotic maps are applied over pixel positions. In a video, there is a huge amount of data present. Permutation of pixels will be a very intensive process. Instead each plane of a frame is divided into macroblocks of size 8x8. A macroblock is smallest quantity in the video we are going to deal with instead of a pixel. One interesting feature of chaotic map is that if the map is applied over a sufficient number of times, the original matrix will be obtained. The map is applied iteratively. If the iterations is a value which is the period of the map, then it will bring back the original information, or if it closer to the period, the information might become easy to decipher. Hence we need to choose the number of iterations carefully before applying the map. It is difficult to predict what the period will be as it depends on the size of the map as well as the seed to the PRNG. Fig 1 and fig 2 shows the relation of chaotic map with iterations based on different value of seed. The graphs are nothing but a plot of cross correlation between the original matrix and its mapped version for a given number of iterations. Greater the value of the cross correlation, greater is the visual similarity between encrypted and original frame. Hence to avoid generation of a map with high correlation (low security), the following precautions are sufficient

(1) The number of iterations is a relatively large prime, 47 in the algorithm. If the value is too high, then the computation time will increase, as well as it could be a period of the map. If it is too small, then the level of randomness will not be sufficient if the map has a large period. If the value is a prime, it reduces the likelihood that it will be a multiple of the period.

(2) A new map, with a new, pseudo-randomly generated seed is used for each plane. This ensures that even if one of the maps has a low level of security, data will get scrambled in some other.

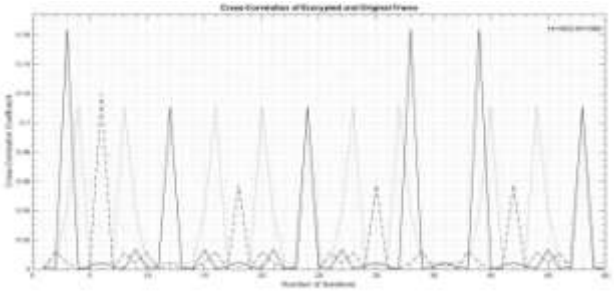


Fig 1: Cross Correlation for frame size1920x1080

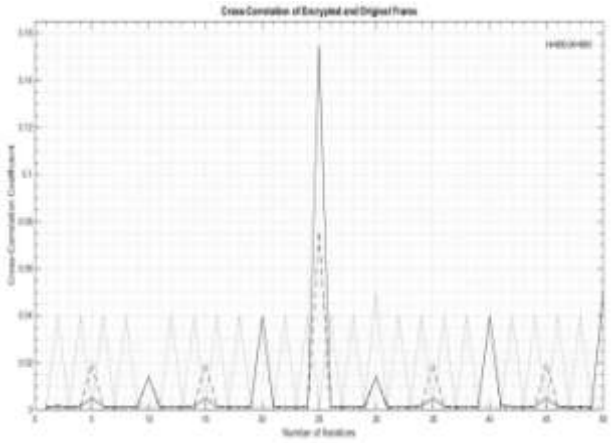


Fig 2: Cross Correlation for frame size 800x600

3.2.1 Applying maps on frames

As mentioned earlier, every single frame of the video file is converted into planes having macroblocks of size 8x8. If the video resolution is$H \times W$, the planes which contain the macroblocks will have resolution $\left(\frac{H}{8}\right) \times \left(\frac{W}{8}\right)$.

The main intention is to apply chaotic maps on three different planes of the video. The front plane, with resolution$\boldsymbol{H \times W}$, side plane(temporal domain), resolution of $\boldsymbol{H \times number\ of\ frames}$ and the top plane (horizontal domain) with resolution $\boldsymbol{W \times number\ of\ frames}$. The mapping in all the three orthogonal planes will introduce sufficient randomness in the video. Macroblocks in different parts of video are mixed and permuted many times to break the correlation between the macroblocks.

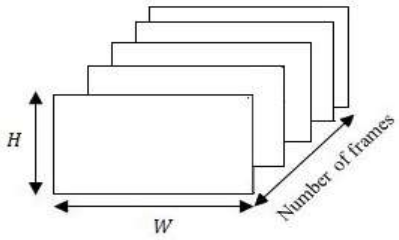


Fig 3: Representation of a video

As seen in fig 4, every single 'R' plane along the dimensions of $H \times W$will be mapped using the 1st map. Similarly the 'G' and 'B' planes will be mapped using the 2nd and 3rd chaotic map respectively.

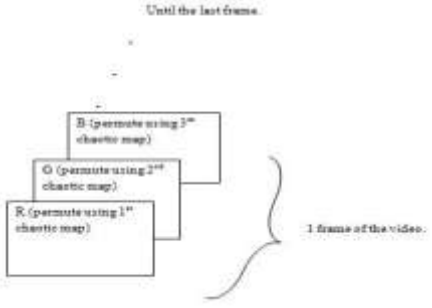


Fig 4: Mapping of RGB planes

Chaotic map of dimensions $H \times number\ of\ frames$ are created using seeds from the original maps. In a similar fashion, macroblocks of the RGB planes are shuffled in the vertical-temporal plane. Chaotic maps of dimensions $W \times number\ of\ frames$ are created using seeds from the previous maps. Macroblocks of RGB planes are permuted in the horizontal-temporal plane. Thus total 9 maps are created for encryption of the whole video. The maps are heavily dependent on the password entered by the user.Simple key expansion algorithms can be used to generate the random values for creating the chaotic maps. A user password is used to generate the 8 random values as well as an Initialization Vector (IV) for the Whirlpool hashof 512 bits.

3.3 Authenticationand Integrity

Authentication is the validity of the message source at the receiver. It guarantees that the message has not been forged. The source is genuine. Integrity refers to maintaining and assuring the correctness and completeness of data over its entire journey.A hash function is a one-way, many-to-one function which maps a digital data of large, arbitrarysize to a data of fixed, small size called as the hash digest of the data. Cryptographic hash functions have a high amount of collision resistance, which means it is computationally difficult to find the original data from the hash. We have chosen two hashes, the Whirlpool hash for data authentication and the XOR hash for data integrity. Whirlpool hash is a relatively new cryptographic hash. It is based on the AES cipher. The maximum input which the hash function can handle is 2^{256} which is sufficient for video data. It produces a small output hash of 512 bits[8].Video data is passed through the function which will produce a 512 bit hash which is finally embedded into the encrypted video. It is only used at the receiver after decryption of the video to verify its authenticity by comparing it to the hash obtained from the encrypted video. Whirlpool hash has been calculated as described in its documentation[9]. XOR is the one of the simplest and fastest hash available. At any intermediate node, the packet is checked for its next address and passed on. If the video is decrypted, it can lead to security breaches. We have very limited time at any node. If the time exceeds the delay requirements, it will reflect on the system performance. Because of these reasons decryption is ruled out at the nodes, and thus XOR hash of encrypted frames is calculated. If there is any change in the content of the video, it will be immediately understood that there was a problem in the current and previous node, and the packet will be dropped.

3.3.1 XOR hash calculation

Instead of calculating the hash of the whole video together, we calculate hash of a few frames and store it. The number of frames$k$ whose hash to be calculated together is found using the formula

$$k = \left\lceil \frac{Number\ of\ frames}{H \times W} \right\rceil$$

We pick the first $k$encrypted frames and XOR all the macroblocks of all the planes. The value is an 8x8 hash. Pick the next k frames and calculate the XOR hash. Continue this procedure till no frames are left. The total number of hashes obtained is given by the following formula

$$Total\ number\ of\ hashes = \frac{Number\ of\ frames}{k}$$

This grouping of frames is done so that an error in the decrypted video can be found in a particular set of *k* frames.

3.3.2 Embedding hash in video

After the hashes have been calculated, it has to be embedded in the encrypted video.To facilitate this, an extra frame is added at the end of the video increasing the number of frames by 1. Since every frame contains 3 planes, we have utilized the 'R' plane for storing the XOR hash and the 'G' plane for the whirlpool hash. Fig 5 shows its diagrammatic representation.

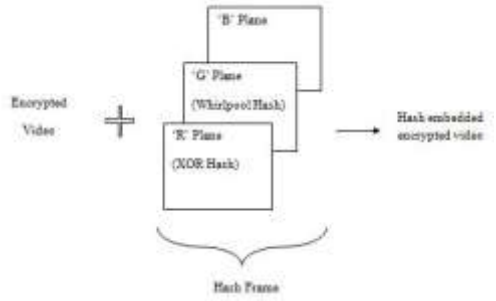


Fig 5: Hash frame embedding

3.4 Key Generation

Key generation is the process of producing keys for encryption and decryption. Since our algorithm heavily depends on seed values, it is necessary to have a mechanism which will generate the same seed for encryption and decryption. A password which is entered by the user has been used to form the seed. The mechanism uses modular exponentiation which forms the backbone of RSA algorithm. The seed values are required for two main purposes-

(1) Random number generation for chaotic maps.
(2) The initialization vector (IV) required for whirlpool hash. It requires a 512 bit IV.It is usually kept as all zeros [9], but using a key based on the password enhances the security of the hash as well as serves the purpose of authentication. Eight permutations of the original substitution box (sBox)[8] of whirlpool areused to generate the IV.

The above explanation shows that the most important aspects of the algorithm, encryption and authentication both are key dependent. If the password is not correct, decryption will be incorrect and authentication will fail.

4. RESULTS AND PERFORMANCE ANALYSIS

All the blocks in the algorithm were implemented in MATLAB for testing and performance analytics. Each algorithmic step was run individually and tested for performance, as well as executed together for obtaining output videos and checking security.

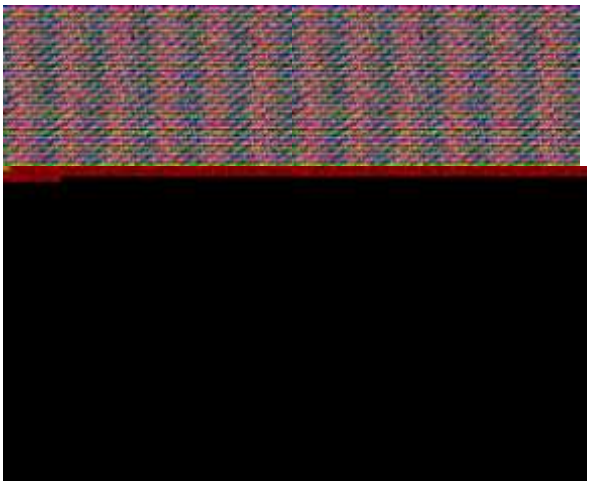
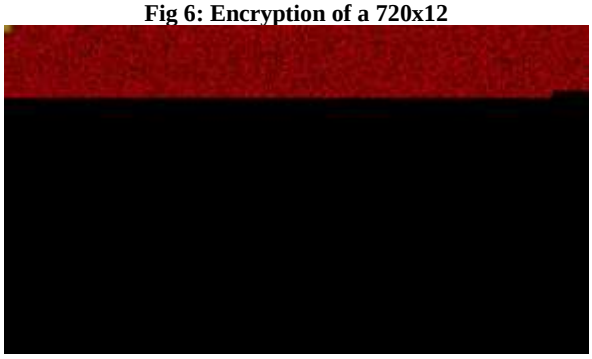


Fig 6: Encryption of a 720x12



80 video and its hash frame

Fig 7: Encryption of a 360x640 video and its hash frame

Our algorithm can be analyzed as a sum of its parts. The main sections of the algorithm are rectangular mapping and hash calculation. Following is a summary of the performance measured. All the codes were run on an Intel Core i7-3610QM running at 2.3 GHz with 6 GB of RAM. Figures 6,7 show the original and encrypted frames of some test videos on which the algorithm was performed. The hash frame is also illustrated showing the XOR and Whirlpool hash.

4.1 Chaotic maps

Generation of the rectangular map depends on two factors, the dimensions of the frame and number of iterations. The graph in fig 8 shows the time for calculation of the map for different sizes and for iterations ranging from 1-500. Time taken is linearly related to the number of iterations. The larger the frame, the greater is the slope of the graph.

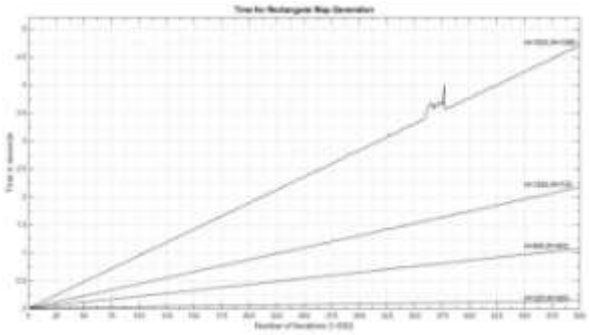


Fig 8: Time for rectangular map generation

4.2 Whirlpool hash

Whirlpool hash is one of the more time consuming areas of encryption. As it is clear fromfig 9, hash calculation is linear with respect to the length of the message. The message size is in terms of 64 byte blocks. The small disturbances in the graph are due to the padding scheme in whirlpool [8].This gives the graph the slight step pattern. Mean time of calculation for the hash is 0.142 ms per macroblock. This is by far the most time consuming step. A tradeoff can be made between security and performance by not using every macroblock in the video for hash calculation. We select a security level between 0.01 and 1. This sets the percentage of macroblocks used for calculation of hash.Random Macroblocks are chosen based on the key, making difficult to predict which blocks are chosen.

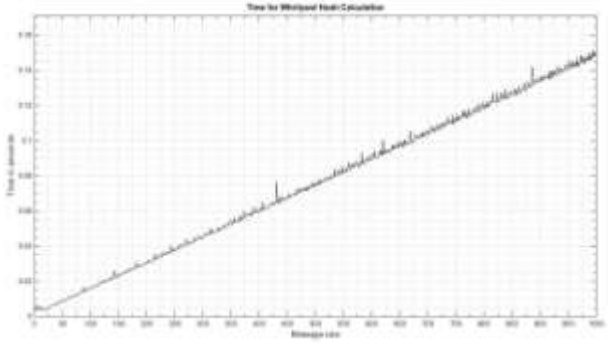


Fig 9: Whirlpool hash execution time

5. SCOPE FOR FUTURE WORK

The method discussed in this paper is in its nascent stages, and has a lot of scope for future work. All algorithms chosen - whether chaotic map generation, mapping of macroblocks, generation of XOR or Whirlpool hash are hardware implementable. Hardware implementation can boost processing speed significantly and also provide better framework for intermediate node check and improved protection against tampering of data.Further improvement in speed can be achieved by parallelization of the different processes.For example chaotic mapping, whirlpool hash calculation and XOR hash calculation can be done in parallel to improve execution speed. The mapping and hash calculation have been performed for entire video. Alternatively, videos can be broken down into chunks of fixed length. Each chunk can be processed in parallel to other chunks. This will allow marked improvement in real time video encryption.Security can also be improved by providing some encryption to macroblock contents. This has not been performed currently due to the excessive increase in processing time. However with correct optimization, additional security can be achieved.

6. CONCLUSION

The encryption method described in this paper makes better use of redundancies specific to video data. It exploits them to encrypt the data faster than existing methods. The complexity involved is very less as compared to other algorithms referred. It has a lot of scope for future work and can be possibly be implemented in real time with suitable buffering and parallelization techniques. It is also important to note, that for the purposes of this paper, all algorithms have been applied to RGB frames. However if compressed video formats like MPEG are to be used, the same principles can be applied to the IPB or any other similar frame structure of the video format.

7. REFERENCES

[1] Deshmukh, P., Kolke, V. 2014. Modified AES based Algorithm for MPEG Video Encryption.

[2] C.NarsimhaRaju, Srinathan, K., Jawahar, C.V. 2008. Real-Time Video Encryption Exploiting Distribution of the DCT coefficients. International Institute of Information Technology, Hyderabad.

[3] Shi, C., Bhargava, B. 1998. An Efficient MPEG Video Encryption Algorithm. Purdue University.

[4] Igorevich, R.R., Yong, H., Dugki Min, Eunmi Choi. 2010. A Study on Multimedia Security Systems in Video Encryption.

[5] Jiri Fridrich. 1998. Symmetric Ciphers Based on Two-Dimensional Chaotic Maps. International Journal of Bifurcation and Chaos.

[6] Salleh, M., Ibrahim, S., Isnin, I.F. 2003. Image Encryption Algorithm Based on Chaotic Mapping.

[7] Zhang, X., Fan, X., Wang, J., Zhao, Z. 2014. A Chaos Based Image Encryption Scheme Using Rectangular Transform and Dependent Substitution.

[8] William Stallings. 2006. The Whirlpool Secure Hash Function.

[9] Baretto, Paulo, S.L.M., Rijmen, V. 2003. The Whirlpool Hashing Function.

[10] Peterson, G. 1997. Arnold's Cat Map