International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:2, No:8, 2008

# A New Approach to Steganography using Sinc-Convolution Method

Ahmad R. Naghsh-Nilchi, and Latifeh Pourmohammadbagher

***Abstract*—**Both image steganography and image encryption have advantages and disadvantages. Steganograhy allows us to hide a desired image containing confidential information in a covered or host image while image encryption is decomposing the desired image to a non-readable, non-comprehended manner. The encryption methods are usually much more robust than the steganographic ones. However, they have a high visibility and would provoke the attackers easily since it usually is obvious from an encrypted image that something is hidden! The combination of steganography and encryption will cover both of their weaknesses and therefore, it increases the security. In this paper an image encryption method based on sinc-convolution along with using an encryption key of 128 bit length is introduced. Then, the encrypted image is covered by a host image using a modified version of JSteg steganography algorithm. This method could be applied to almost all image formats including TIF, BMP, GIF and JPEG. The experiment results show that our method is able to hide a desired image with high security and low visibility.

***Keywords*—**Sinc Approximation, Image Encryption, Sinc-convolution, Image Steganography, JSTEG.

## I. Introduction

STEGANOGRAPHY was originally formalized by Simmons as the prisoners problem [1]. Steganography is, in fact, the art of invisible communication. Its aim is to hide the very presence of communication by embedding messages into innocuous-looking cover objects. To accommodate a secret message (in an image form, for example) in a host digital image, the original cover (host) image is slightly modified by the embedding steganography algorithm [13,14]. The most important required characteristic for a steganographic system is *its undetectability*: stego images should be statistically indistinguishable from cover images. In other words, there should be no artifacts in the stego image that could be detected by an attacker with probability better than random guessing, given the full knowledge of the embedding algorithm, including the statistical properties of the source of cover images, except for the stego key (Kerckhoffs' principle). While this characteristic is desired, there is no available steganographic system that reached this aim.

Manuscript received June 25, 2006.

A. R. Naghsh-Nilchi is an assistant professor of computer engineering at the University of Isfahan, Isfahan, 81744, Iran (phone: +98-311-7932662; fax: +98-311-7932670; e-mail: nilchi@yahoo.com).

L. Pourmohammadbagher is a graduate student at the University of Isfahan, Computer Engineering Department, Isfahan, 81744, Iran (e-mail: mroshanzamir@eng .ui.ac.ir).

Common methods of concealing data in digital images include:

- Least significant bit insertion: This is a very popular method which the LSB of each byte in the image is used to store the secret data. The resulting changes are too small to be recognized by the human eye. The disadvantage of this technique is that since it uses each pixel in an image, a lossless compression format like bmp or gif has to be used for the image. If lossy compression is used, some of the hidden information might be lost [2, 4].
- Masking and filtering: These methods hide information in a manner similar to paper watermarks. This can be done, for example, by modifying the luminance of parts of the image. It does change the visible properties of an image, but if done with care the distortion is barely discernable [6].
- Transformations: Discrete cosine transformation (DCT) is one such method. DCT is used by the JPEG compression algorithm to transform successive 8 * 8 pixel blocks of the image, into 64 DCT coefficients each. Steganography tools can use the LSB of the quantized DCT coefficient to hide information (JSteg method). In addition to DCT, images can be processed with fast Fourier transformation as well as wavelet transformation. Other image properties such as luminance can also be manipulated [7].
- Spread Spectrum Image Steganography: Methods based on this technology encode the desired message to be hidden. For encoding, a wide band pseudorandom noise generator is used to create a spreading sequence. Then, a modulation scheme is used to spread the narrowband spectrum of the message with the spreading sequence, thereby composing the embedded signal which is tehn enters into interleave and spatial spreader. The inner leaver may also use a key to dictate the interleaving algorithm. This signal is now combined with the cover image to produce the stego image, which has been appropriately quantized to preserve the initial dynamic range of the cover image. The stego image is then transmitted to the recipient of the message [15].

Image encryption methods encode the desired message (in image format), such that it would be changed to some non-readable, non-comprehended form, usually look like two dimensional random noise signals. These methods are usually involved with an encryption key which is used to decode the image with a relatively high security.

While it is a difficult task for an attacker to detect the right key, it is easy to suspect an encrypted image carrying some confidential information (form how it looks!). This

allows the intruder to intercept the transmission of the image, for example, by destroying it.

On the other hand, the security level of steganography methods is a great challenge. There is no steganographic methods that could completely meet this challenge, today.

Therefore, a joint image encryption and steganographic method could make the desired image message much safer. In this paper, we introduced a joint stego-encryption method to meet such challenge.

In this method, a desired image is first encrypted using an algorithm based on sinc-convolution introduced in[11,12]. Then a JPEG-JSTEG steganographic method is used to hide the encrypted image in a host image used as a stego cover.

In the followings, we first describe the sinc-convolution method used to encrypt a desired image. Then, a popular steganographic method called JSTEG which is used to hide the encrypted image is briefly described. Then, our algorithm for both encoding and decoding procedure is discussed. This follows some implementation results.

## II. Sinc-Convolution Encryption

### A. Sinc and Sinc-Convolution Approximation

Sinc function is defined as follows [9]:

$$\operatorname{sinc} z = \frac{\sin(\pi z)}{\pi z} \tag{1}$$

Then, by selection a constant value, h, which is defined as

$$h = \left(\frac{\pi d}{\beta N}\right)^{\frac{1}{2}} \tag{2}$$

and then selecting an appropriate mapping function, $\phi$, and from it, finding the sinc sampling points defined by

$$z_j = \phi^{-1}(jh) \quad , j \in Z \tag{3}$$

we are approximating an convolution integral function, p(x), at M+N+1 sampling points, by computing:

$$\begin{aligned}
&\gamma_j = Sinc\left(\frac{\phi - jh}{h}\right) \quad , j = -M,...,N \\
&\omega_j = \gamma_j, \quad j = -M+1,...,N-1 \\
&\omega_{-M} = \frac{1}{1+\rho} - \sum_{j=-M+1}^{N} \frac{1}{1+e^{jh}}\gamma_j \\
&\omega_N = \frac{\rho}{1+\rho} - \sum_{j=-M}^{N} \frac{e^{jh}}{1+e^{jh}}\gamma_j \\
&p(x) \approx \sum_{i=-M}^{N} p_i \omega_i(x)
\end{aligned} \tag{4}$$

To estimate the function p(x) by sinc algorithm, First, the position and integer value of N will be determined, Next, $M = \left[\beta N / \alpha\right]$ will be calculated, so that $0 < \alpha \le 1$ and $0 < \beta \le 1$ m=M+N+1 will give the number of sinc points. Assuming that the values of the function p(x) at sinc points are available and are displayed as $\rho_j (j = -M,...,N)$ and assuming that $\rho = e^{\phi}$ [10, 11].

Sinc-convolution uses sinc-approximation method to approximate multi-dimensional integral equations in the form of convolution. Sinc-convolution method is also used to solve multi-dimensional partial differential equations by means of an indefinite convolution procedure combined with Sinc approximation. Using this procedure, we introduced an encryption method described as follows.

### B. Image Encryption using Sinc-Convolution

In this method, an image is divided into the appropriate number of parts and each part will be convolved by the function φ. The function will be selected so that it will match Sinc-convolution algorithm. Note that a digital image could be described as a two-dimensional function. So, we use the two dimensional mode of convolution integral to encrypt the image with a selected two-dimension function. Let us consider g(x,y) to define our image, then a function p(x,y) is defined as the two-dimensional convolution integral between the image, g(x,y) and a function f(x,y) [10]:

$$p(x,y) = \iint_B f(x-t, y-\tau) g(t,\tau) dt d\tau \tag{5}$$

In this formula, *g(x,y)* is an image function which is to be encrypted and *f(x,y)* is the encryption function and the result of *p(x,y)* will be the encryption of the sub-area B of the image. The Eq. (5) is a suitable integral equation matching the sinc-convolution algorithm introduced in [9,11,12].

Note that to solve convolution integral using the numeral sinc-convolution method, it is necessary to have the two-dimensional Laplace transform of the function *f(x,y)*. Table I shows some suitable functions with their Laplace transform. The combination of the parameters included in each of these functions is used as the key for encryption [12].

To encrypt an image, first, the image will be compressed and encrypted by the key. To see the image, the image should be decrypted first and reconstructed next. First we set

$$m_x = M_x + N_x + 1, m_y = M_y + N_y + 1$$

and we define the sinc points by

$$Z_l^{(j)} = \varphi^{-1}(lhj), l = -M_j,...,N_j; j = 1,2$$

next, we determine matrix *g* so that:

$$\left[g_{i,j}\right] = \left[g(Z_i^{(x)}, Z_j^{(x)})\right], i = -M_x,...,N_x, j = -M_y,...,N_y \tag{6}$$

The more the number of sinc points, the better the quality of the reconstructed image would be, and therefore, the capacity of the matrix $[g_{ij}]$ will increase.

The needed Laplace transform of f(x,y) is defined by:

$$F(S^{(1)},S^{(2)}) = \int_0^\infty \int_0^\infty f(x,y) e^{-x/S^{(1)} - x/S^{(2)}} dxdy \quad (7)$$

As one may observe, the Laplace transform variables in Eq. (7) is inverted ("*1/s*" and "*1/v*" instead of usual variable "*s*" and "*v*"). Now, the encryption algorithm is as follows [11].

$$P(a,b) = \int_0^N \int_0^M g(x,y) f(a-x, b-y) dxdy \quad (8)$$

1. $U_{ij} = [B(Z_i^{(1)}, Z_j^{(2)})]$, where the sinc points are defined as $Z_i^{(k)}, k = 1,2, i = -M_k, ..., N_k$ .
2. $A_i = h_i I_{m_i} D(\frac{1}{\varphi_i'}),\ i = 1,2$ so that,

$$D(\frac{1}{\varphi_i'}) = diag(\frac{1}{\varphi'_{(2M)}}, ..., \frac{1}{\varphi'_{(2N)}}),$$

$$I_{mk} = \frac{1}{2} + \int_0^k \frac{\sin(\pi z)}{\pi z} dz, k \in z \quad (9)$$

3. $A_i = X_i S_i X_i^{-1},\ i = 1,2$
4. $H_{ij} = X_1^{-1} U_{ij},\ K_{ij} = X_2^{-1} H_{ij}$
5. $T_{ij} = E(S_i^{(1)}, S_j^{(2)}) K_{ij}$
6. $Q_{ij} = X_2 T_{ij},\ P_{ij} = X_1 Q_{ij}$, $P_{ij}$ is the encrypted image of this algorithm [10].

### C. Encryption Key

For more security, images are divided into several parts as in Fig. 3 and each section is encrypted with a function. "Header" that includes the important information of the image has been divided into smaller sections [ ].

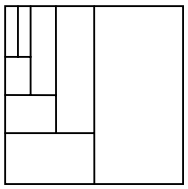


Fig. 3 Encryption Key parts

The key used for encryption is selected to be 128 bits long and is composed of two parts:

1. Encryption information, including the number of image parts, the number of each function and its parameters.
2. Security information including the secret information.

Elements of the encryption key:

1. The first 8 bits to determine the number of image parts (at most 8 bits and at least 1 bit).
2. The second 8 to 64 bits depends on the number of image parts and determines the function number through which each part of the image is encrypted (Table I).
3. The third 16 to 48 bits determine the parameters of the encryption function (at most 6 parameters and at least 2 parameters).
4. The remaining bits for the key security.

After encryption of the desired message in a form of an image, we use JSTEG steganography algorithm to hide it in a cover (host) image. The following section, describes the JSTEG algorithm.

## III. JSTEG

JPEG-JSteg algorithm is a typical steganographic algorithm using JPEG file as carrier-image. JSteg is first proposed by D. Upham [13]. In this method, after quantization of DCT coefficients, JPEG-JSteg replaces the least significant bits (LSB) of the quantized DCT coefficients by the secret message bits. The embedding mechanism skips all coefficients with the values 0 or 1[16].

The steganographic tool Jsteg[4] embeds messages in lossy compressed JPEG files. It has a high capacity—e.g., 12 % of the steganogram's size—and, it is immune against visual attacks. However, a statistical attack discovers changes made by Jsteg [5, 8].

In this paper we present a new methodology using both sinc-convolution encryption algorithm and JSteg steganography to encrypt and hide a desired message in a form of an image with a very high level of security. Figs. 1 and 2 show proposed algorithm for both encoding and decoding process.

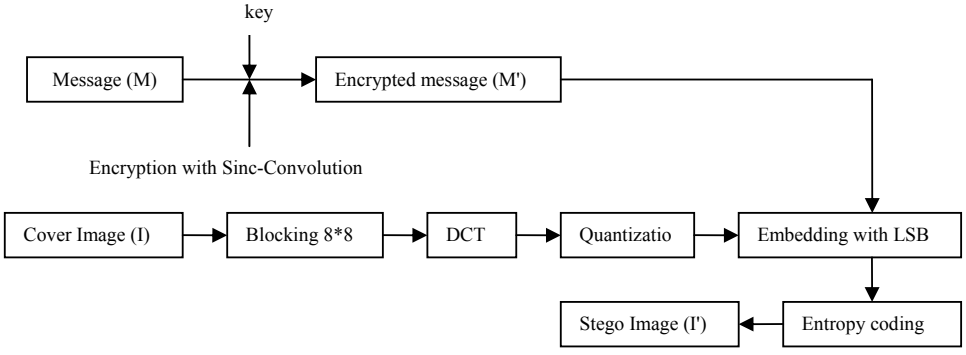



Fig. 1 Encoder

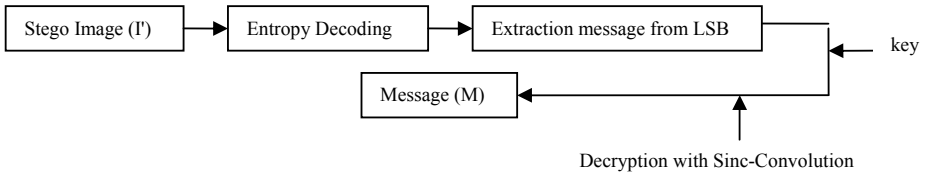



Fig. 2 Decoder

TABLE I
LAPLACE TRANSFORM

| No. | Function | Laplace transform | Function parameters |
|---|---|---|---|
| 1 | $\exp(-\alpha x)$ | $s/(1+\alpha s)$ | $\alpha,\beta$ |
| 2 | $[1/(\alpha-\beta)][\exp(-\alpha x)-\exp(-\beta x)]$ | $s^2/(\alpha\beta s^2+(\alpha+\beta)s+1)$ | $\alpha,\beta,\lambda,\gamma$ |
| 3 | $\sin(\alpha x)$ | $As^2/(1+\alpha^2 s^2)$ | $A,\alpha$ |
| 4 | $\cos(\alpha x)$ | $s/(1+\alpha^2 s^2)$ | $A,\alpha$ |
| 5 | $\exp(-\alpha x)\sin(\beta x)$ | $\beta s^2/[(\alpha^2+\beta^2)s^2+2\alpha s+1]$ | $\alpha,\beta,\gamma,\lambda$ |
| 6 | $\exp(-\alpha x)\cos(\beta x)$ | $(\alpha s^2+s)/[(\alpha^2+\beta^2)s^2+2\alpha s+1]$ | $\alpha,\beta,\gamma,\lambda$ |
| 7 | $\exp(-\alpha x)\{\cos(\beta x)+[(\lambda-\alpha)/\beta]\sin(\beta x)\}$ | $(\lambda s^2+s)/[(\alpha^2+\beta^2)s^2+2\alpha s+1]$ | $\alpha,\beta,\gamma,\lambda,\eta,\theta$ |

## IV. ENCODING PROCESS

### A. Decompression

In Sinc method, having the matrix [$g_{ij}$] the primitive image can almost be reconstructed through it's mapping against the neighbor pixels. If the number of Sinc points are enough (table 1), the lost points f the primitive matrix will n not be clear.

### B. Image Decryption

The receiver will receive $P_{ij}$'s as an encrypted image. The decryption algorithm steps are against the encryption algorithm i.e. the encryption algorithm will receive $P_{ij}$ and the key and will give $U_{ij}$ that is the compressed image. Decryption algorithm is as followings [10]:

1. $$A_i = h_i I_{m_i} D(\frac{1}{\varphi_i'}) = X_i S_i X_i^{-1},\ i = 1,2$$
2. $$Q_{ij} = X_1^{-1} P_{ij},\ T_{ij} = X_2^{-1} Q_{ij} \qquad (10)$$
3. $$K_{ij} = E^{-1}(S_i^{(1)}, S_j^{(2)}) T_{ij}$$
4. $H_{ij} = X_2 K_{ij},\ U_{ij} = X_1 H_{ij}$ , $U_{ij}$ is the primitive compressed image at sinc points.

## V. IMPLEMENTATION RESULTS

In the above mentioned method, Steganography capacity depends on the number of quantized DCT coefficients which are not equal to 0 and 1. Therefore this method offers a limited capacity and its capacity could be estimated (about 12.8% of image size). It is indicated for 4 covered images (Fig. 4) in Table II.

This method is secure against visual attacks, but a statistical attack can discover changes which are created by JSteg algorithm.

The kind of selected image will be more effective in getting better results after steganography. Actually this method will be broken because of the changes created in histogram of DCT coefficients, therefore by using of encryption algorithm, errors of JSteg will be eliminated and encrypted data will be hidden. "Cameraman" image encrypted with a special key is indicated in Fig. 5.

Pic.1 Pic. 2 Pic. 3 Pic. 4

Fig. 4 Covered picture

TABLE II
CAPACITY OF EACH IMAGE IN JSTEG METHOD

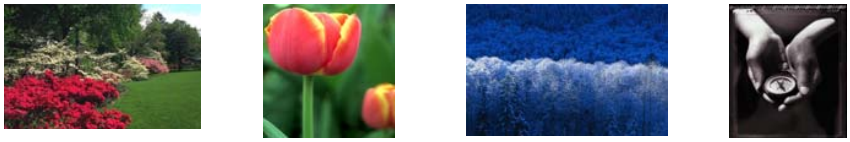
| Embedding capacity(Q=75,KB) | Embedding capacity (Q=90,KB) | Image size(KB) | Image dimensions | Image |
|---|---|---|---|---|
| 110165 | 50744 | 6701 | 640*480 | Image 1 |
| 10480 | 30344 | 172 | 487*730 | Image 2 |
| 110387 | 160997 | 103 | 800*600 | Image 3 |
| 10331 | 20806 | 20 | 389*480 | Image 4 |

encrypted image of cameraman with 170484126441922911562 1

original image of cameraman

Fig. 5 Original image and encrypted image of "cameraman"

Fig. 6 will represent image resulted by esteganography of encrypted "cameraman" image in covered images.



PSNR=35.54 Pic. 1 PSNR=41.95 Pic. 2 PSNR=33.61 Pic. 3 PSNR=39.7 Pic. 4

Fig. 6 Stego images with embedded data with different PSNRs

Extracted and decrypted data image with true key is indicated in Fig. 7.a and extracted and decrypted image with false key is indicated in Fig. 7.b.

7.a. Decryption with function key 570484126441922911562 1

7.b. Decryption with function key 170484126441922911562 1

Fig. 7 Encryption of "cameraman" image with different functions

## VI. CONCLUSION

Experimental results show that the proposed joint image encryption and steganography method could provide more security for today's need of secure communication. Adding the 128 bit key made the algorithm even more difficult to be broken. These advantages should make the algorithm a feasible image hiding methodology.

## REFERENCES

[1] G. Simmons, "*The Prisoners' Problem and the Subliminal Channel*", CRYPTO83 Advances in Cryptology, pp. 51{67, Aug 22 -24, 1984}.
[2] R. B. Wolfgang and E. J. Delp., "*A Watermark For Digital Images*", Proceedings of the IEEE Inter-national Conference on Image Processing, Lausanne, Switzerland, pp. 219–222, September 1996.
[3] B. Schneier, "*Applied Cryptography*", New York, John Wiley & Sons, 1996, pp. 9.
[4] R. van Schyndel, A. Tirkel and C. Osborne, "*A Digital Watermark*", Proceedings of the IEEE International Conference on Image Processing, pp. 86–90, 1994.
[5] Andreas Westfeld, Andreas Pfitzmann, "*Attacks on Steganographic System*", third international workshop, LNCS 1768, Springer-Verlag Berlin Heidelberg 2000. pp. 61–76.
[6] Robert Krenn, "*Steganography and Steganalysis*", chapter 1.
[7] A.Westfeld, "*F5—A Steganographic Algorithm, high Capacity Despite Better Steganalysis*", Technische Universit¨at Dresden, Institute for System Architecture, D-01062 Dresden, Germany.
[8] Peter Wayner, "*Disappearing Cryptography*", 2nd Edition, by Elsevier Science (USA), 2002.
[9] F. Stenger, A.R. Naghsh-Nilchi, J. Niebsch, R. Ramlau, "*Sampling Methods for Approximation Method for Approximate Solution of PDE*", Vol. 313, pp. 199-249 of Contemporary Mathematics, American Mathematical Society, 2002.
[10] F. Stenger, "*Numerical Methods Based on Sinc and Analytic Functions*", Springer Verlag,, 1993.
[11] Naghsh-Nilchi, A. R., Khanche, M. E., "*Image Compression and Encryption using Sinc-Convolution*", Sheik Bahaee Engineering Science Journal, Vol. 3, No. 1, 2003 (in Farsi).
[12] Haidari, B., "*Using 128 bit Key for Image Encryption using Sinc-Convolution"*, M.S. thesis, Azad University of Najaf-Abad, 2005 (in Farsi).
[13] G. Goth, "*Steganalysis Gets Past the Hype*", IEEE Computer Society, Vol. 6, No. 4, April 2005
[14] *B. Koukuntla, R. Kumar, A. Lach, "Steganography* ", CSCIE2005.
[15] Lisa M. Marvel,Charles T. Retter , "*A Methodology For Data Hiding Using Images*", IEEE 1998.
[16] N. Rasouli , " *Image Steganography with JPEG Format* " , MSc Thesis , Islamic Azad University Unit of Najaf Abad, 2004.