Data Hiding Method with Modified Interpolation and Key Based Sudoku

Dr. Shailja Shukla
Prof. and HOD of CSE
Jabalpur Engineering College, Jabalpur (M.P.)
shailja270@gmail.com

Dr. Veerendra Kumar
Director of Tech. Education Bhopal (M.P.)
prof.veerendra.kumar@gmail.com

Dr, Vandana Roy
Asst. Prof.
HCET, Jabalpur (M.P.)
vandana.roy20@gmail.com

Ragya Kori
Student of M.E. control system,
Jabalpur Engineering College, Jabalpur (M.P.)
ragyakori@gmail.com

Abstract: This work intends to give an overview of image steganography, its uses & techniques. Paper work is an implementation of Image Steganography of same plaintext and the implementation of a system that designs a unique Sudoku for Sudoku based data hiding. Higher security is the main concern of this paper. The system is based on a hybrid algorithm that applies the technique of Sudoku generation and steganography to offer different security features to images transmitted between entities in internet. Based on the proposed algorithm, the authenticity and integrity of the transmitted images can be verified either in the spatial domain or in the encrypted domain or in both domains. The work is implemented on MATLAB design and simulation tool. Paper work uses modified Sudoku based data protection and PSNR is very high in present paper work.

Keywords: Peak Signal to Noise Ratio (PSNR), Mean Square Error (MSE), Bit Per Pixel (BPP), Sudoku

I-INTRODUCTION

The paper work is the Implementation of steganography tools for hiding more data or information which may be text or another image files. Transform techniques have been used for identifying appropriate data. Available methods in area of steganography have been presented:-

1. Initially LSB substitution in image was popular [15],[16]
2. Different schemes for LBS substitution developed [14]
3. Key based LSB substitution developed [13]
4. Multiple layer for steganography developed like steganography with steganography [12]
5. DCT based image analysis for finding out area where data can be hidden appropriately [3]
6. DWT based image analysis for finding out area where data can be hidden appropriately & to find better method than DCT based image analysis [7][6]
7. Sudoku based data hiding method in cover image [1],[2]

II-METHODOLOGY

Proposed design resolves the problems of available works with significant modification done in interpolation method and in Sudoku generation method. 8 digit public and private key have been used in the proposed work, and association of both the keys develops one 8 digit Sudoku Key. Proposed work is a new design which uses Sudoku, developed with 8 digits Sudoku Key provided by user. This method develops a unique Sudoku solution for each different key, hence the concept of unique Sudoku based on combination of public and private key makes proposed method robust against external intrusion. Proposed work is also using an interpolation and new pixel generation method with the help of four neighboring pixels instead of two pixels as was used by Chin-Chen Chang et al [1], four pixels interpolation develops more accurate new pixel than two pixels interpolation. The remaining processes of hiding are similar as used by Chin-Chen Chang et al [1].

2.1 DATA HIDING METHOD ADOPTED:

Fig. 1 presents the flow diagram of the proposed work, explained in following steps-

Step 1: Input an 8 digit decimal public key, the key can be any value of 8 digit and must be pass when data hiding required.

Step 2: perform logical XOR between 8 digit public and private key which was already in the steganography module. The output key is our new Sudoku Key.

Step 3: develop a unique t1 to t9 digits out of 8 digits

Sudoku key, t1 to t9 can be any digits between 1 to 9 but all t1 to t9 must be different from each other.

Step 4: Assign the values of t1 to t9 into a fixed bottom up approach Sudoku problem. Total 23 unknown values of Sudoku problem has been assigned using t1 to t9 digits.

Step 5: Solve the bottom up Sudoku problem by using Sudoku Rules, proposed method uses the method of solving Sudoku by Ahmed Abdulkarim Almuhrij et al [27] from University of Manchester.

.

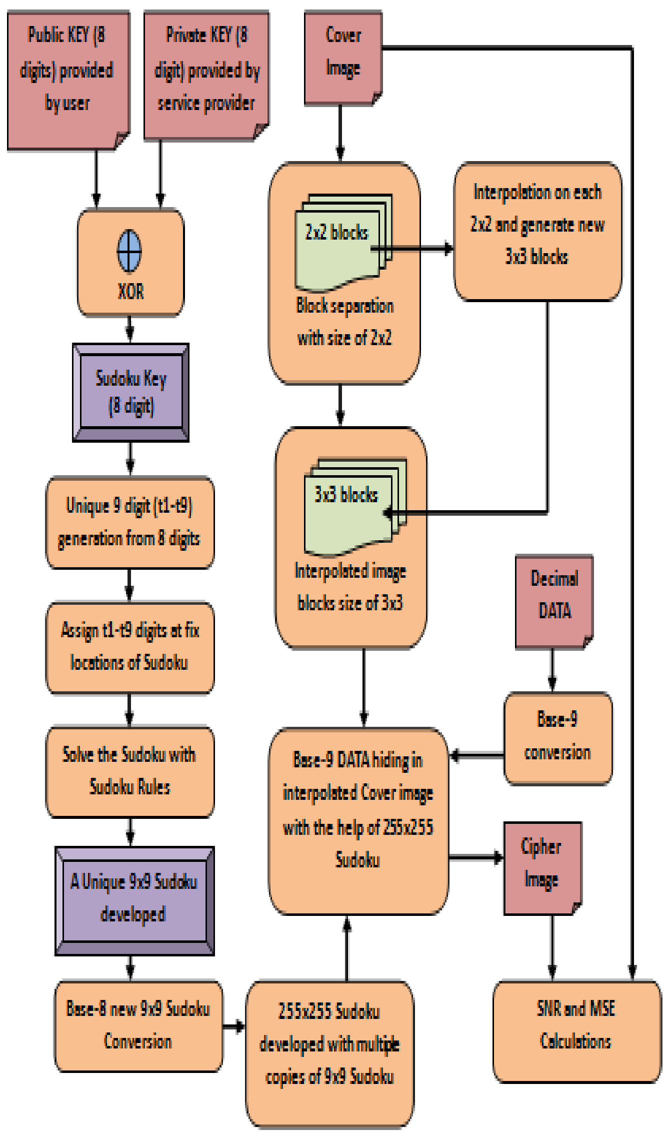

Figure 1 Data hiding Flow diagram

Step 6: convert the base-9 Sudoku into base-8 Sudoku, because we can hide digits 0-8 only.

Step 7: develop a 255x255 Sudoku matrix with help of base-8 developed Sudoku, the size of 255x255 is fixed because any pixel of image cannot be greater than 255.

Step 8: input the cover image and isolate it in 2x2 blocks

Step 9: perform interpolation on each 2x2 blocks of cover image and develop new 3x3 interpolated blocks, total 4 pixels was available in 2x2 block and total 9 pixels have been developed in 3x3 block, hence we have 5 new pixels. These 5 new pixels can be modified according to the data which is to be hidden in the cover image.

The role of interpolation is that original pixels (4 pixel of 2x2 block) must not change and new 5 interpolated pixels can be modified with small amount, hence overall image modification is less as required in steganography.

Step 10: input the data which may be character or any number, first convert it into its ASCII values then develop a single string of decimal digits.

Step 11: convert the number into base-9

Step 12: start hiding the data in cover image with the help of 255x255 Sudoku matrix. Actually the original digit is not hidden in the cover image, the digits of data simply modify the interpolated pixels of cover image. Each 2x2 block of interpolated image can hide 3 digits.

2.2 PROPOSED ALGORITHM FOR DATA HIDING : Input a 8 digit public key, let its 8 digits are K1, K2….K8

KEY=K1K2K3K4K5K6K7K8

Shift row for making Key complex as also done in AES

NK=K2K3K4K5K6K7K8K1

Perform logical XOR of 'NK' with private key 'PK' for public Key encryption

MK=NK xor PK

Now develop sub-keys from MK

MK1=99999999-MK

MK2=88888888-MK

MK3=77777777-MK

MK4=66666666-MK

MK5=55555555-MK

MK6=44444444-MK

MK7=33333333-MK

MK8=22222222-MK

On the other hand Let Sud is a 9x9 matrix where the positions t1,t2….t9 are fixed

$$Sud = \begin{matrix} t1 & UN & UN & UN & UN & t2 & UN & t3 & UN \\ UN & t4 & UN & UN & t5 & UN & UN & UN & t6 \\ UN & UN & t7 & t8 & UN & UN & t9 & UN & UN \\ UN & UN & t1 & t2 & UN & UN & t3 & UN & UN \\ UN & t5 & UN & UN & t4 & UN & UN & UN & t7 \\ t6 & UN & UN & UN & UN & t8 & UN & UN & UN \\ t9 & UN & UN & UN & UN & UN & UN & t1 & UN \\ UN & t2 & UN & UN & UN & UN & UN & UN & t3 \\ UN & UN & t4 & UN & UN & UN & t5 & UN & UN \end{matrix}$$

Following method is used to generate t1, t2…..t9, the values of t1, t2 ….t9 can never be same.

Initially i=1 and j=1 and p=1

‘i’ can range maximum from 1 to 23

‘j’ can range maximum from 1 to 9

‘p’ can range maximum from 1 to 8

$$
\begin{aligned}
& For\ i \\
& = 1{:}\,23 \\
& For\ j \\
& = 1{:}\,8 \\
& \quad if\ \left(MK(j)_{dp} \neq (t_{1:i-1}, t_{1:i-2}, t_{1:i-3} \ldots\ldots t_1\right) \\
& \qquad\qquad t_i = MK(j)_{dp} \\
& \quad elseif\ \left(MK(j)_{d(p+1)} \neq (t_{1:i-1}, t_{1:i-2}, t_{1:i-3} \ldots\ldots t_1\right) \\
& \qquad\qquad t_i = MK(j)_{d(p+1)} \\
& \quad elseif\ \left(MK(j)_{d(p+2)} \neq (t_{1:i-1}, t_{1:i-2}, t_{1:i-3} \ldots\ldots t_1\right) \\
& \qquad\qquad t_i = MK(j)_{d(p+2)} \\
& \qquad\qquad \vdots \\
& \qquad\qquad \vdots \\
& \qquad\qquad else \\
& \qquad\qquad t_i = MK(j)_{d(p+7)} \\
& \quad end \\
& end
\end{aligned}
$$

Ones the values of t1 to t9 computed, based on public key provided by user and private key by the service provider, a unique Sudoku need to be designed according to computed values.

The remaining values of UN in Sudoku matrix ’sud’ will be computed using Sudoku solver according to the Sudoku rules :

Any row cannot have repeated numbers 1 to 9

Any column cannot have repeated numbers 1 to 9

Any 3x3 square (corning and central)matrix numbers cannot have repeated numbers 1 to 9

A MATLAB define function names SudokuSolver.m can solve this Sudoku and it can compute all the unknown values using above rules, the complete Sudoku is represented below where t1-t9 are initially computed with the help of the KEY and U1-U59 are computed with the help of SodokuSolver.m and Sudoku rules.

This Sudoku is completely defined by the key, as the key changes the complete Sudoku will also change, and this method will provide a good avalanche which is small change in input KEY cause major changes in output Sudoku, hence the system can be considered a chaotic system.

$$
Sud1 = \begin{matrix}
t1 & U1 & U2 & U3 & U4 & t2 & U5 & t3 & U6 \\
U7 & t4 & U8 & U9 & t5 & U10 & U11 & U12 & t6 \\
U13 & U14 & t7 & t8 & U15 & U16 & t9 & U17 & U18 \\
U19 & U20 & t10 & t11 & U21 & U22 & t12 & U23 & U24 \\
U25 & t13 & U26 & U27 & t14 & U28 & U29 & U30 & t15 \\
t16 & U31 & U32 & U33 & U34 & t17 & U35 & U36 & U37 \\
t18 & U38 & U39 & U40 & U41 & U42 & U43 & t19 & U44 \\
U45 & t20 & U46 & U47 & U48 & U49 & U50 & U51 & t21 \\
U52 & U53 & t22 & U54 & U55 & U56 & t23 & U57 & U58
\end{matrix}
$$

‘SudM’ contains digits from 0 to 8 only and can be developed as: SudM=Sud1 -1

Now make copies of SudM such that a matrix of 255x255 is generated.

SudM	SudM	....	SudM	
SudM	SudM	....	SudM	
..	..	..	...	...
..	..	..	...	...
..	..	..	...	...
SudM	SudM	....	SudM	

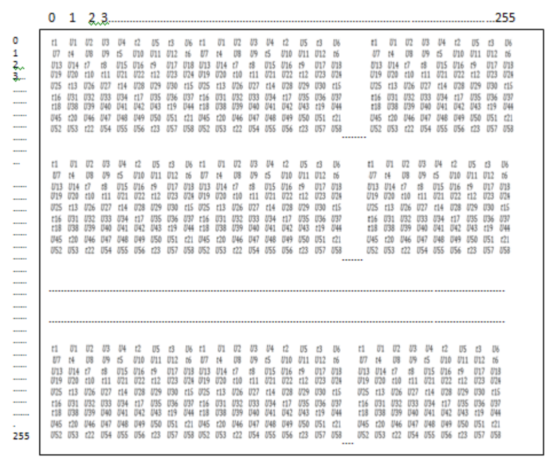

Figure 2: 255x255 Sudoku matrix

Input the cover image and convert it into matrix form which shows the pixels of the image. Let one part of the cover image as below

$$
img = \begin{matrix} r1_1 & r1_2 & r1_3 \\ r2_1 & r2_2 & r2_3 \\ r3_1 & r3_2 & r3_3 \end{matrix}
$$

Perform interpolation using the formula below

$$
\boldsymbol{rx_{ni} = \left[rx_i + \left(\frac{rx_i + rx_{i+1}}{2}\right)\right] \big/ 2} \qquad \boldsymbol{where\ x\ is\ constant}
$$

$$
\boldsymbol{cx_{ni} = \left[ri_x + \left(\frac{ri_x + r(i+1)_x}{2}\right)\right] \big/ 2} \qquad \boldsymbol{where\ x\ is\ constant}
$$

$$rcx_{ni} = \left\{ \left[rx_{ni} + \left(\frac{rx_{ni}+rx_{n(i+1)}}{2} \right) \right] \Big/ 2 + \left[cx_{ni} + \left(\frac{cx_{ni}+cx_{n(i+1)}}{2} \right) \right] \Big/ 2 \right\} \Big/ 2$$

where x is constant

$$img1 = \begin{matrix} r1_1 & r1_{n1} & r1_2 & r1_{n2} & r1_3 & r1_{n3} \\ c1_{n1} & rc1_{n1} & c1_{n2} & rc1_{n2} & c1_{n3} & rc1_{n3} \\ r2_1 & r2_{n1} & r2_2 & r2_{n2} & r2_3 & r2_{n3} \\ c2_{n1} & rc2_{n1} & c2_{n2} & rc2_{n2} & c2_{n3} & rc2_{n3} \\ r3_1 & r3_{n1} & r3_2 & r3_{n2} & r3_3 & r3_{n3} \\ c3_{n1} & rc3_{n1} & c3_{n2} & rc3_{n2} & c3_{n3} & rc3_{n3} \end{matrix}$$

Input data ‘D’ and convert it into ASCII format and then to base-9 formats

ND=(D)9

Figure 3 selection of row according to pixel in 255x255 Sudoku matrix

Now from the interpolated cover image ‘img1’ select $(r1_1, r1_{n1})$ position from Sudoku, obtained in step 8 and then consider 4 pixel ahead and 4 pixels back from the position $(r1_1, r1_{n1})$.

If the nine digits of Sudoku are [t14 U28 U29 U30 t15 U25 t13 U26 U27], search for the first digit of new data ‘ND(1)’ and detect its position $(r1_1, Y_{n1})$

Replace the searched new position $(r1_1, Y_{n1})$ with pixel of $(r1_1, r1_{n1})$ of interpolated image, repeat this process until all the digits of the new data (ND) are not replaced,

$$g1 = \begin{matrix} r1_1 & Y_{n1} & r1_2 & Y_{n2} & r1_3 & Y_{n3} \\ Y_{n4} & rc1_{n1} & Y_{n5} & rc1_{n2} & Y_{n6} & rc1_{n3} \\ r2_1 & Y_{n7} & r2_2 & Y_{n8} & r2_3 & Y_{n9} \\ Y_{n10} & rc2_{n1} & Y_{n11} & rc2_{n2} & Y_{n12} & rc2_{n3} \\ r3_1 & Y_{n13} & r3_2 & Y_{n14} & r3_3 & Y_{n15} \\ Y_{n16} & rc3_{n1} & Y_{n17} & rc3_{n2} & Y_{n18} & rc3_{n3} \end{matrix}$$

The whole idea is that we are not making any changes in the original information of the cover image, the pixel values of interpolated pixel gets modified which was generated using interpolation formula, and making slight change in this pixel does not affect quality on the image significantly.

2.3 PROPOSED METHOD FOR DATA EXTRACTION

Step 1: Input an 8 digit decimal public key, the key can be any value of 8 digit and must be pass when data extraction required and it must be same as entered at the time of data hiding.

Step 2: Perform logical XOR between 8 digit public and private key which is already defined in the steganography module. The output key is known as Sudoku Key.

Step 3: Develop a unique t1 to t9 digits out of 8 digits Sudoku key, the t1 to t9 can be any value between 1 to 9 but all must be different from each other.

Step 4: Assign the values of t1 to t9 into a fixed bottom up approach Sudoku problem. Total 23 unknown values of Sudoku problem must be assigned using t1 to t9 digits.

Step 5: Solve the bottom up Sudoku problem with using Sudoku Rules, proposed method uses the method of solving Sudoku by Ahmed Abdulkarim Almuhrij et al [27] from University of Manchester.

Step 6: Convert the base-9 Sudoku into base-8 Sudoku because we can hide digits 0-8 only.

Step 7: Develop a 255x255 Sudoku matrix with the help of base-8 developed Sudoku, the size of 255x255 is fixed because any pixel of image cannot be greater than 255.

Step 8: Input the cipher image and isolate its 2x2 blocks.

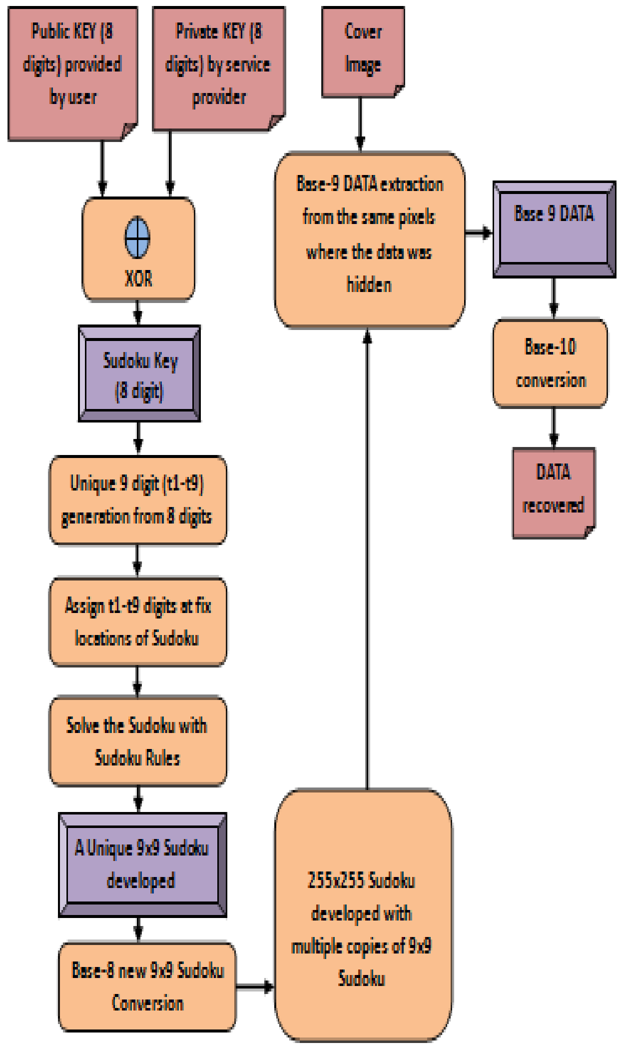

Figure 4 Data Extraction flow diagram

Step 9: With the help of 255x255 Sudoku matrix generate 3 decimal digits from each 2x2 block. The digits must be put as a sting of data.

Step 10: Convert the number into base-10.

Step 11: Develop ASCII from the digit string and then convert ASCII into Characters.

2.4 ALGORITHM FOR DATA EXTRACTION

The generation of 255x255 Sudoku is same as was in data hiding.

Input the cover image and convert into matrix form which shows the pixels of the image. Let one part of the cover image as below:-

$$\text{img}\begin{bmatrix} P11 & P12 & \cdots\cdots & P1(N-1) & P1N \\ P21 & P22 & \cdots\cdots & P2(N-1) & P2N \\ P31 & P32 & & P3(N-1) & P3N \\ P41 & P42 & \cdots\cdots & P4(N-1) & P4N \\ \vdots & & & & \vdots \\ P(N-1)1 & P(N-1)2 & & P(N-1) & P(N-1)N \\ PN1 & PN2 & \cdots & PN(N-1) & PNN \end{bmatrix}$$

Isolate 2x2 block, let say:

$$\text{Blk}=\begin{matrix} P11 & P12 \\ P21 & P22 \end{matrix}$$

Now from the cipher image 'img1' select (P11, P12), (P11, P21) and (P11, P22) position and search corresponding values in 255x255 Sudoku, the values in Sudoku will be the data in base-9 form.

Collect values from Sudoku according to positions (P11, P12), (P11, P21) and (P11, P22) and develop a string of base-9 data values.

Convert base-9 data string into decimal data string and decimal data string into ASCII and then convert ASCII into alphanumeric characters, these characters will be the final extracted DATA.

III-RESULTS

MSE, SNR and BPP are the results parameters, these parameters will help in the observation of results and comparison of proposed work with others.

$$MSE = \frac{\left\{\sum_{i=1}^{rw}\sum_{j=1}^{cl}[P_{cipher}(i,j) - P_{cover}(i,j)]\right\}^2}{rw * cl}$$

$$PSNR = 10 * log_{10} * \frac{256^2}{MSE}$$

$$BPP = \frac{8 * Bd}{Bi}$$

Pcipher is the pixel of cipher image
Pcover is pixel of cover image
Rw is number of row in cipher image
Cl is the number of column in cipher image
Bi number of bits in the cipher image
Bd number of bits in data

BPP: Bit Per Pixel is the number of bits that can be hidden inside a pixel, The capacity of data which can be hidden inside the proposed work can be explained with a test image let say image of 'Lena' with 512x512 pixels

512x512=2, 62, 144 pixels

Total 256x256 number of 2x2(4 pixel block) can be developed with 512x512 image

256x256x2x2=2,62,144 pixels

And after interpolation 2x2 block convert into 3x3 block means total

256x256x3x3=5,89,824 pixels

And as it's a color image with 3 frames total pixels can be

256x256x3x3x3= 17,69,472 pixels

As proposed method can hide three digits (12 bit) in each 2x2 block of cover image

17,69,472/4 = 4,42,386 total 2x2 blocks available

And each block can hide 12 bits hence 4,42,386 x 12 = 53,08,632 bits can be hidden in interpolated image size of 256x256x3x3x3 pixels and each pixel is of 8bit hence 256x256x3x3x3x8= 1,41,55,776 bits in interpolated image.

BPP= 53,08,632 x 8 / 141,55,776 = 3.000122

Hence the maximum BPP observed for the proposed method is 3.000122.

Number of bits	PSNR	MSE x 10^{-11}
2000	54.53	5.41
4000	53.47	6.91
8000	53.01	7.68
16000	52.15	9.37
32000	50.95	12.37
64000	49.84	15.95
128000	48.83	20.13

Table 1 PSNR observation for various size of data

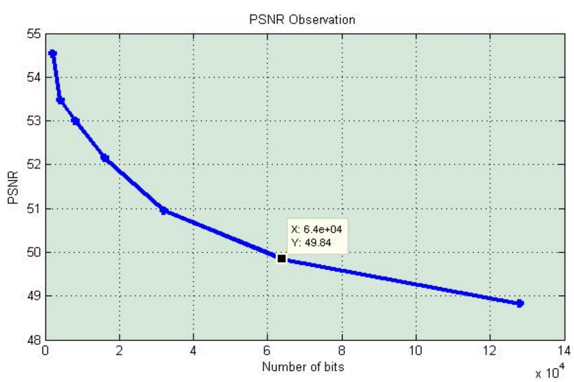

Figure 6 PSNR measured for different numbers of bits

3.1 COMPARATIVE RESULTS

PSNR Results observed for the 512x512 Lena Standard image			
2000 bits of data		**1,20,000 bits of data**	
Chang-Tsun Li et al [13]	**Proposed Work**	**Thai-Son Nguyen et al [2]**	**Proposed Work**
46.25	54.53	48.67	48.83

Table 2 PSNR Comparison with available work

Sr. NO.	AUTHOR	Results
1.	Fan Li et al [3]	PSNR observed is 52.99 when BPP taken as 0.063 for 512x512 Lena image
2.	Chin-Chen Chang et al [1]	2.23 bpp hidden in Lena Image obtain PSNR of 26.86
3.	Proposed work	PSNR is 32.33 for 3.000122 BPP hidden in Lena image of 512x512 pixels

Table 3 BPP Comparison with available work

From the comparative results it can be observed that proposed work PSNR is better than available works and also the proposed work can hide more bits per pixel than available work.

IV-CONCLUSION

The original objective for paper work was to develop an optimized technique for hiding data inside cover image also to reduce amount for data in the channel while stenograph data transmission which has been achieved. A new 8 digit decimal number based unique Sudoku developed for enhancing the robustness of the work and also a modified interpolation is been developed for maintaining good quality of image after hiding the data. Problem with steganography is that it needs lots of data (image) for sending few small amount for data, so proposed work is a good solution for this problem it can be confidently said that, because we have achieved very good PSNR. Proposed design can also be used for secure communication in cloud services. The data bits are embedded in random into the cover-image pixels, instead of it sequential embedding can be done to improve the security of the system in future.

REFERENCES

1. [1] C. C. Chang, T. S. Nguyen, Y. Liu, A reversible data hiding scheme for image interpolation based on reference matrix, 2017 5th International Workshop on Biometrics and Forensics (IWBF) , DOI: 10.1109/IWBF.2017.7935098, Coventry, UK IEEE Xplore: 29 May 2017
2. [2] T. S. Nguyen , C. C. Chang, A reversible data hiding scheme based on the Sudoku technique, http://dx.doi.org/10.1016/j.displa.2015. 10.003-0141-9382/2015 Elsevier, Science Direct Displays 39 (2015) 109–116
3. [3] F. Li, Q. Mao, and C. C. Chang, A Reversible Data Hiding Scheme Based on IWT and the Sudoku Method, International Journal of Network Security, Vol.18, No.3, PP.410-419, May 2016 410
4. [4] S. Chakraborty, Prof. S. K. Bandyopadhyay, Steganography Method Based On Data Embedding By Sudoku Solution Matrix, International Journal of Engineering Science Invention ISSN (Online): 2319 – 6734, ISSN (Print): 2319 – 6726 www.ijesi.org Volume 2 Issue 7, July 2013, PP.36-42
5. [5] R. N. Louise, Creating Sudoku Puzzles, Control # 2883, February 19, 2008 Page 1 of 15 Control # 2883, math.swansea.ac.uk/staff/jhg/papers/MCM2008-Ruth-Nick-Louise.pdf
6. [6] S. Ijeri, S. Pujeri, S. B. Usha, Image Steganography using Sudoku Puzzle for Secured Data Transmission, International Journal of Computer Applications (0975 – 888) Volume 48– No.17, June 2012 31
7. [7] The Duc Kieu, Zhi-Hui Wang, Chin-Chen Chang, and Ming-Chu Li, A Sudoku Based Wet Paper Hiding Scheme, International Journal of Smart Home, Vol.3, No.2, April, 2009
8. [8] Ayushi, A Symmetric Key Cryptographic Algorithm, 2010 International Journal of Computer Applications (0975 - 8887), Volume 1 – No. 15
9. [9] S. Dixit, A. Gaikwad, S. Gaikwad, S. A. Shanwad, Public Key Cryptography Based Lossless and Reversible Data Hiding in Encrypted Images, Volume 6 Issue No. 4 DOI 10.4010/2016.822, ISSN 2321 3361 © 2016 IJESC
10. [10] Y. T. Lin, C. M. Wang, W. S. Chen, F. P. Lin, and W. Lin, Novel Data Hiding Algorithm for High Dynamic Range Images, IEEE TRANSACTIONS ON MULTIMEDIA, VOL. 19, NO. 1, JANUARY 2017
11. [11] A. A. Haj, H. A. Nabi, Digital Image Security Based on Data Hiding and Cryptography, 2017 3rd International Conference on Information Management, 978-1-5090-6306-2/17/2017 IEEE
12. [12] S. Rawal, Advanced Encryption Standard (AES) and It's Working, International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056 Volume: 03 Issue: 08 | Aug-2016
13. [13] C. C. Chang, C. T. Li, Reversible Data Hiding in JPEG Images Based on Adjustable Padding, 2017 5th International Workshop on Biometrics and Forensics (IWBF), DOI: 10.1109/IWBF.2017.7935083, IEEE Xplore: 29 May 2017
14. [14] D. Maglott, J. Ostell, K. D. Pruitt and T. Tatusova, Entrez Gene: gene-centered information at NCBI, D52–D57 Nucleic Acids Research, 2011, Vol. 39, Database issue Published online 28 November 2010, doi:10.1093/nar/gkq1237
15. [15] A. K. Maji, S. Roy, R. K. Pal, A Novel Algorithmic approach for solving Sudoku puzzle in Guessed Free Manner, EUROPEAN ACADEMIC RESEARCH, VOL. I, ISSUE 6/ SEPEMBER 2013 ISSN 2286-4822
16. [16] A. U. Islam, F. Khalid, M. Shah, Z. Khan, T. Mahmood, A. Khan, U. Ali, M. Naeem, An Improved Image Steganography Technique based on MSB using Bit Differencing, INTECH 2016, 978-1-5090-2000-3/16/2016 IEEE
17. [17] M. Wilhelm, I. Martinovic, E. Uzun, and J. B. Schmitt, SUDOKU: Secure and Usable Deployment of Keys on Wireless Sensors, INCCON 2010, 978-1-4244-8915-2/10/2010 IEEE
18. [18] B. S. Xiwei, S. Y. WU Yilong, Y. G. YANG, A New Algorithm for Generating Unique-solution Sudoku, Fourth International Conference on Natural Computation, 978-0-7695-3304-9/08-2008 IEEE, DOI: 10.1109/ICNC.2008.788
19. [19] M. Iwata, K. Miyake, A. Shiozaki, Digital steganography utilizing features of JPEG images, IEICE Trans. Fundam. Electron. Commun. Comput. Sci. E87-A(2004) 929–936.
20. [20] J. Mielikainen, LSB matching revisited, IEEE Signal Process. Lett. 13 (2006) 285–287.
21. [21] I. J. Cox, J. Kilian, F. T. Leighton, T. Shamoon, Secure spread spectrum watermarking for multimedia, IEEE Trans. Image Process. 6 (1997) 1673–1687.
22. [22] J. Tian, Reversible data hiding using difference expansion, IEEE Trans. Circuits Syst. Video Technol. 13 (2003) 890–896.
23. [23] D. M. Thodi, J. J. Rodriguez, Expansion embedding techniques for reversible watermarking, IEEE Trans. Image Process. 16 (2007) 721–730.
24. [24] Z. Ni, Y.Q. Shi, N. Ansari, W. Su, Reversible data hiding, IEEE Trans. Circuits Syst. Video Technol. 16 (2006) 354–362.
25. [25] H. Luo, F. X. Yu, H. Chen, Z. L. Huang, H. Li, P. H. Wang, Reversible data hiding based on block median preservation, Inf. Sci. 181 (2011) 308–328.
26. [26] Y. C. Li, C. M. Yeh, C. C. Chang, Data hiding based on the similarity between neighboring pixels with reversibility, Digital Signal Process. 20 (2010).
27. [27] A. B. Almuhrij, SUDOKU GAME, Dr. David Lester, 2011, University of Manchester