International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 06 | June 2020 www.irjet.net p-ISSN: 2395-0072

Secure Reversible Image Data Hiding using Parametric Binary Tree Labeling Algorithm

Marteena P.M

Student, Dept. of Dual Degree Computer Applications, Sree Narayana Guru Institute of Science and Technology N.Paravur, Kerala, India

---

Abstract - *Today, the demand of internet has made the transmission of digital media much easier and faster. Open nature of internet, risks of illegitimate accessing and unauthorized tempering with transmitted data is increased day by day. Protection of secret information from unauthorized users during a public network has become a crucial issue. Data hiding is one among the foremost demanding techniques to guard the safety of digital media. We have proposed reversible data hiding techniques for digital images. In this technique, cover image is split into block equal size.*

Extracted secret text is similar to original secret text Here I use parametric Binary Tree Labeling (PBTL). So that the embedding rate can be significantly improved. This paper first introduces a parametric binary tree labeling scheme (PBTL) to label image pixels in two different categories. Using PBTL, a knowledge embedding method (PBTL-DE) is proposed to embed secret data to a picture by exploiting spatial redundancy within small image blocks. We then apply PBTL-DE into the encrypted domain and propose a PBTL based reversible data hiding method in encrypted images (PBTLRDHEI). PBTL-RDHEI may be a separable and reversible method that both the first image and secret data are often recovered and extracted losslessly and independently. Experiment results and analysis show that PBTL-RDHEI is in a position to realize a mean embedding rate as large as 1.752 bpp and a couple of .003 bpp when block size is about to 2 × 2 and three × 3, respectively.

Key Words Reversible data hiding, encrypted images, parametric binary tree labeling scheme, privacy protection

1. INTRODUCTION

Cloud computing has recently reached popularity and developed into a serious trend in IT. We perform such a scientific review of cloud computing and explain the technical challenges facing during this paper. In Public cloud the "Pay per use" model is employed. In private cloud, the computing service is distributed for one society. In Hybrid cloud, the computing services is consumed both the private cloud service and public cloud service. Cloud computing has three types of services. Software as a Service (SaaS):In this model, a complete application is offered to the customer, as a service on demand. A single instance of the service runs on the cloud & multiple end users are serviced. On the customers‟ side, there's no need for upfront investment in servers or software licenses, while for the provider, the prices are lowered, since only one application needs to be hosted & maintained. Platform as a Service (Paas):Here, a layer of software, or development environment is encapsulated & offered as a service, upon which other higher levels of service are often built. The customer has the liberty to create his own applications, which run on the provider's infrastructure. To meet manageability and scalability requirements of the applications, PaaS providers offer a predefined combination of OS and application servers, like LAMP platform (Linux, Apache, MySql and PHP),Infrastructure as a Service (Iaas):IaaS provides basic storage and computing capabilities as standardized services over the network. Servers, storage systems, networking equipment, data centre space etc. are pooled and made available to handle workloads. The customer would typically deploy his own software on the infrastructure. Some common examples are Amazon, GoGrid, 3 Tera, etc.It is also known as Hardware as a Service(HaaS).Public clouds are owned and operated by third parties; they deliver superior economies of scale to customers, because the infrastructure costs are spread among a mixture of users, giving each individual client a beautiful low-cost, "Pay-as-you-go" model. All customers share an equivalent infrastructure pool with limited configuration, security protections, and availability variances. These are managed and supported by the cloud provider. Private clouds are built exclusively for one enterprise. They aim to deal with concerns on data security and offer greater control, which is usually lacking during a public cloud. Hybrid Clouds combine both public and private cloud models. With a Hybrid Cloud, service providers can utilize 3rd party Cloud Providers during a full or partial manner thus increasing the pliability of computing.

Reversible image data hiding (RIDH) may be a special category of knowledge hiding technique, which ensures perfect reconstruction of the duvet image upon the extraction of the embedded message. The reversibility makes such image data hiding approach particularly attractive within the critical scenarios, e.g., military and remote sensing, medical images sharing, law forensics and copyright authentication, where high fidelity of the reconstructed cover image is required.

In this paper, I propose an RDHEI method using parametric binary tree labeling scheme (PBTL-RDHEI). It is a VRBE

method that keeps spatial correlations within small encrypted image blocks, so that secret data embedding can be accomplished by exploiting the spatial redundancy from the encrypted image. Different from the methods in that use the traditional RDH method to embed secret data, we adopt the PBTL based reversible data embedding so that the embedding rate can be significantly improved. The contributions of this paper are summarized as follows:

1) We propose a parametric binary tree labeling scheme (PBTL) to label pixels in two different categories. Selecting different settings of parameters, PBTL will provide different pixel labeling strategies.

2) Using PBTL, we propose a data embedding algorithm (PBTL-DE). It exploits spatial redundancy in small image blocks and embeds secret data into cover images using pixel labeling and bit replacement. Different from the traditional data embedding methods that embed secret data by modifying the plaintext cover image pixel values in an imperceptive way, PBTL-DE is designed for encrypted images. Thus, the significant changes to pixel values are acceptable.

3) Based on PBTL-DE algorithm, we further propose a PBTL-based RDHEI method (PBTL-RDHEI). Simulation results of applying PBTL-RDHEI to 1000 randomly selected test images demonstrate that PBTL-RDHEI is able to achieve a mean embedding rate as large as 1.752 bpp and 2.003 bpp when block size is set to 2×2 and 3 × 3, respectively.

2. EXISTING SYSTEM

The majority of the existing RIDH algorithms are designed over the plaintext domain, namely, the message bits are embedded into the original, un-encrypted images. The early works mainly utilized the lossless compression algorithm to compress certain image features, in order to vacate room for message embedding. Histogram shifting (HS)-based technique is another class of approach achieving better embedding performance through shifting the histogram of some image features. The latest difference expansion (DE)-based schemes and the improved prediction error expansion (PEE)-based strategies were shown to be able to offer the state-of-the-art capacity distortion performance.

2.1. Disadvantages

As the source coding with side information at the decoder requires a feedback channel, this scheme would face severe challenges in many practical scenarios, e.g., secure remote sensing, where the feedback channel could be very costly. The embedding capacity of this type of method is rather limited and the incurred distortion on the watermarked image is severe.

3. PROPOSED SYSTEM

In this work, we propose an encrypted-domain RIDH scheme by specifically taking the above-mentioned design preferences into consideration. The proposed technique embeds message through a public key modulation mechanism, and performs data extraction by exploiting the statistical distinguishability of encrypted and non-encrypted image blocks. Since the decoding of the message bits and the original image is tied together, our proposed technique belongs to the category of non-separable RIDH solutions Compared with the state-of-the-arts, the proposed approach provides higher embedding capacity, and is able to achieve perfect reconstruction of the original image as well as the embedded message bits. Extensive experimental results on test images validate the superior performance of our scheme.

3.1. Advantages

Enabling us to jointly decode the embedded message and the original image signal perfectly. It provide higher security and have higher embedding capacity. Also it providing higher security to data and image.

4. ARCHITECTURAL DESIGN

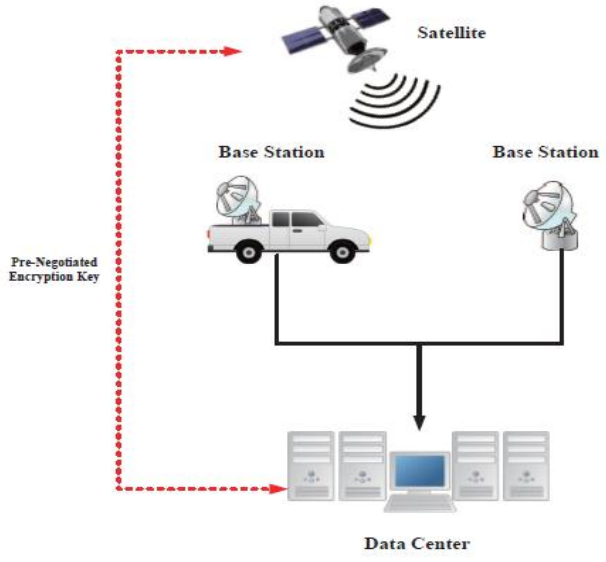


Fig -4.1: Architectural Design

5. PARAMETRIC BINARY TREE LABELING SCHEME

In this section, we propose a parametric binary tree labeling scheme (PBTL). it's designed to label pixels in two different categories, namely G1 and G2. For pixels with 8-bit depth, we use α and β bits of code to label pixels in C1 and C2, respectively, where 1 6 α, β 6 7.

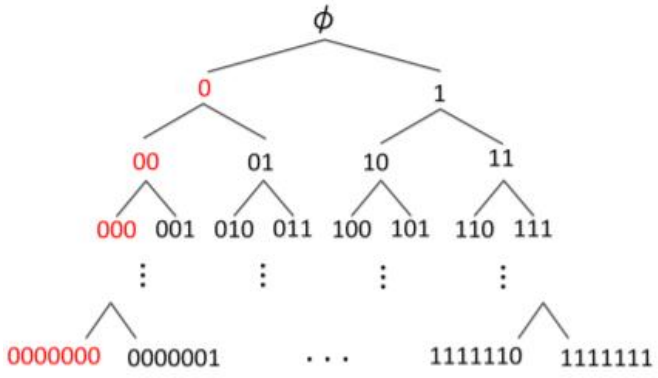
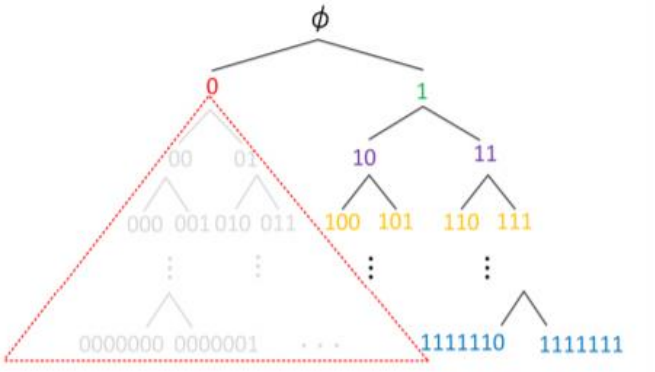


Fig-5.1: Distribution of binary codes based on a full binary tree.



(a)

$\beta=1$	G_2	G_1
$\alpha=1$	0	1

$\beta=1$	G_2	G_1	
$\alpha=2$	0	10	11

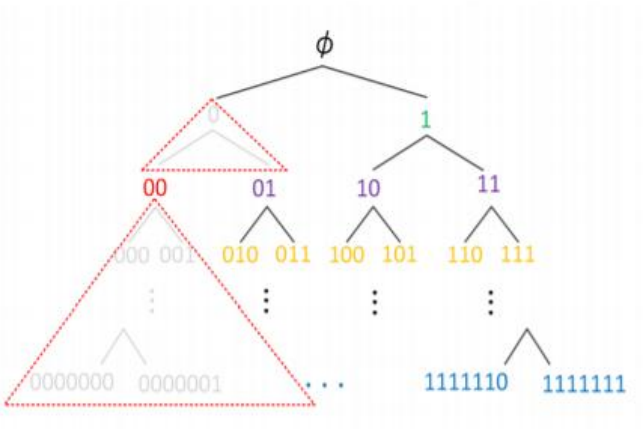
$\beta=1$	G_2	G_1			
$\alpha=3$	0	100	101	110	111

• • • • • •

$\beta=1$	G_2	G_1		
$\alpha=7$	0	1000000	.....	1111111

(b)

Fig 5.2: Fig. 2: Example of labeling bits selection when β = 1 and α = 1 to 7.



(c)

$\beta=2$	G_2	G_1
$\alpha=1$	00	1

$\beta=2$	G_2	G_1		
$\alpha=2$	00	01	10	11

$\beta=2$	G_2	G_1					
$\alpha=3$	00	010	011	100	101	110	111

• • • • • •

$\beta=2$	G_2	G_1		
$\alpha=7$	00	0100000	.....	1111111

(d)

Fig 5.3: Example of labeling bits selection when β = 2 and α = 1 to 7.

To better explain our idea, we use a full binary tree structure, as shown in Fig. 5.1, for instance the distribution of binary labeling bits. As are often seen, the binary tree has 7 layers of child node, and the i th layer contains 2 i nodes, where i = 1, 2, ..., 7. First of all, given a parameter β, we use the code in the primary node of the β th layer to label pixels in C2. Thus, '0...0 |{z} β ' is adopted. For C2, all pixels are labeled by an equivalent labeling bits '0...0 |{z} β '. For C1, consistent with the known value β and another given parameter α, pixels are classified into na sub-categories, where nα is calculated by Eq. (1). nα = (2 α − 1, if α 6 β (2β − 1) ∗ 2 α−β , otherwise (1) For pixels during a sub-category, we use an equivalent α-bit binary code to label them, and for pixels in several sub-categories, different α-bit binary codes are applied. Thus, nα different α-bit binary codes are utilized to label nα sub-categories, respectively. Next, we analyze the content of those nα binary codes from the subsequent three aspects. (1) When α = β, as shown in Fig. 5.1, the first node of the β

th layer '0...0 |{z} β ' is chosen to label pixels in C2, and the remaining nα = 2α − 1 nodes of binary codes within the same layer are utilized to label pixels in nα sub-categories of C1, respectively. When α < β, for every of the α th layer, we ignore the primary node and use the remaining nα = 2α−1 nodes of binary codes to label pixels in nα sub-categories of C1. The illustrative examples are often found in Figs. 5.2 and 5.3. When α > β, for every of the α th layer, only the binary codes that aren't derived from the node '0...0 |{z} β ' are selected to label pixels in nα sub-categories of C1. Figs. 5.2-5.3 show the examples of labeling bits selection when α = 1 to 7, β equals to 1, 2 and 3, respectively. As are often seen, for instance, in Fig. 5.2, all binary codes that derived from '0' are ignored and the remaining binary codes in α th layer are kept.

6. DATA EMBEDDING

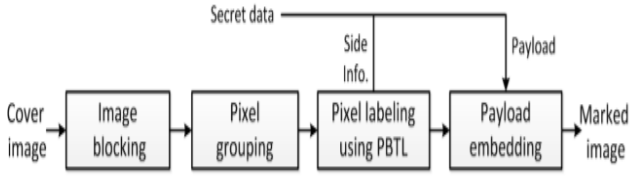


Fig 6.1: Framework of PBTL-DE.

The data embedding consist of 5 steps.

Step 1: Image blocking

For an 8-bits depth original image I with a size of M × N, we first divide it into variety of s×s non-overlapped small blocks. for instance , the block size is about to 2 × 2 or 3 × 3.

Step 2: Pixel grouping

For all pixels in I, separate them into four sets, namely: reference pixel (Pr), special pixel (Ps), embeddable pixel (Pe) and non-embeddable pixel (Pn). Here, Pr consists of nr pixels that selected by user-defined rules. for instance , we select the primary pixel (or center pixel) of every 2 × 2 (or 3 × 3) block to make Pr. These pixels are going to be kept unmodified during data embedding phase. Ps contains one pixel which can be utilized to store some parameters. Any pixel except in Pr can be selected to be Ps. Without loss of generality, we choose one pixel within the first block to be Ps. Thus, for every block except for the primary one, one reference pixel is corresponding to 3 (for 2×2) or 8 (for 3×3) non-reference pixels; otherwise, one reference pixel is like 2 (for 2 × 2) or 7 (for 3 × 3) non-reference pixels. Then, for every of the remaining (MN − nr − 1) pixels Ii(i = 1, 2, ..., MN − nr − 1), we calculate its difference value ei by ei = Ii − I ref i (2) where I ref i ∈ Pr is that the corresponding reference pixel of Ii . If ei satisfied the subsequent condition, the pixel Ii belongs to Pe; otherwise, it's in set Pn. d−nα 2 e 6 ei 6 b nα − 1 2 c (3) where na may be a positive integer, d∗e and b∗c are the ceil and floor operations, respectively. Here, Pe and Pn contain ne and nn pixels, respectively, where pixels in Pe are often utilized to embed secret data while Pn can't . Thus, MN = nr +ne + nn+1. Fig. 6 shows an example of pixel grouping when block size is 2 × 2. After obtaining the difference set e = {ei} MN−nr−1 i=1 , we can obtain its histogram h(e) by h(e) = #{1 6 i 6 MN − nr − 1 : ei = e}, ∀e ∈ Z (4) where # is that the cardinal of a group. Thanks to the spatial correlations of pixels within an equivalent block, the histograms of e form sort of a Laplace distribution with location parameter equals to 0. As shown in Eq. (3), so as to realize a better embedding rate, we use the pixel whose difference value falls into the nα center bins of histogram h(e) to embed secret data.

Step 3: Pixel labeling using PBTL

Because the pixel locations of Pr and Ps are pre-defined, they can be easily distinguished, we only got to label the pixels in Pn and Pe. Given two parameters α and β, we use the binary codes generated by PBTL to label pixels in Pn and Pe, respectively. for instance , for every pixel in Pn, a β-bit code '0...0 |{z} β ' is adopted to label it by bit replacement, and the remaining (8−β) bits are kept unmodified. For pixels in Pe, they will be classified into nα sub-categories according to different values of e. Thus, nα different α-bit binary codes are utilized to label pixels in each sub-category, respectively.

Step 5: Payload embedding

The payload contains three parts: the first 8 bits of pixel in Ps, the replaced original β bits of every pixel in Pn, and the secret data. After pixel labeling, the remaining (8−α) bits of every pixel in Pe are reserved to embed payload bits by bit replacement. Thus, totally (8−α)ne bits of the payload are often successfully embedded. The parameters α and β are important for data extraction and image recovery, thus, they have to be stored as well. Since 1 6 α, β 6 7, they will be successfully stored by 8 bits in Ps by bit replacement. Therefore, the marked image is generated, and therefore the detailed procedures of RDH using PBTL are provided in Algorithm 1. Then, we will calculate the effective embedding rate rα,β (.bpp) under different settings of parameters α and β by rα,β = (8 − α)ne − βnn − 8 MN (5) which is like rα,β = (8 − α) Pvr i=vl h(i) − β(Pvl−1 j=−255 h(j) + P255 k=vr+1 h(k)) − 8 MN (6) where vl = d−nα 2 e and vr = b nα−1 2 c. Then the utmost embedding rate rmax (.bpp) are often calculated by rmax = max{rα,β} 7 α,β=1 (7) An illustrative example of pixel labeling and data embedding when α = β = 2 is shown in Fig. 7. As are often seen, '00' is employed to label pixels in Pn, and therefore the remaining 6 bits are kept unmodified. consistent with Eq. (1), nα = 3. Thus, '01', '10' and '11' are applied to label pixels in Pe when the difference value e adequate to -1, 0 and 1, respectively.

ALGORITHM: PBTL-DE

Input: Original image I, Secret data M, parameters α and β.
1: Divide I into equal size non-overlapping blocks and classify pixels into four groups Pr, Ps, Pe and Pn.
2: Construct the payload P, where it consists of the secret data M, the first β bits of each pixel in Pn and 8 bits in Ps.
3: for each pixel in Pe do
4: According to the difference value e, reconstruct the pixel by replacing α labeling bits and (8 − α) payload bits.
5: end for
6: for each pixel in Pn do
7: Replace its first β bits by '0...0 |{z} β ' and keep the remain (8 − β) bits unmodified.
8: end for
9: Convert α and β into binary bits and store them into Ps by bit Replacement.
Output: Marked image ˆI

IMPLIMENTATION

The creation of the designed system takes place within the implementation phase. Development phase overview, preparation of implementation, computer virus development, development phase report and overview. It also performs activities like writing, testing, debugging and documenting the programs. This is often to review the performance of the system and to guage against standard or criteria. A study is conducted for measuring the performance of the system against pre-defined requirements. Database design forms a crucial a part of every project. The management of knowledge involves both the definition of structure for the storage of data and provision of mechanisms for manipulation of data. The database system must provide safety for the knowledge stored; despite system crashes or attempts of unauthorized access the database utilized in this project is MYSQL.

CONCLUSION

In this paper, I first proposed a parametric binary tree labeling scheme (PBTL). Using PBTL, then proposed a data embedding method (PBTL-DE) and further applied it into the encrypted images application (PBTL-RDHEI). The PBTL-DE is restricted designed for encrypted-domain based application, because it significantly changed the pixel values in the image. PBTL-RDHEI may be a full reversible method that both the secret data and original image are often extracted without any error. Experiment results and comparisons have shown that the PBTL-RDHEI significantly improved the embedding rate. Security analysis has demonstrated the robustness of PBTLRDHEI in withstanding brute-force and know/chosen-plaintext attacks.

Design a secure reversible image data hiding (RIDH) scheme operated over the encrypted domain. We suggest a public key modulation mechanism, which allows us to embed the data via simple XOR operations, without the need of accessing the secret encryption key. At the decoder side, we propose to use a powerful two-class SVM classifier to discriminate encrypted and non-encrypted image patches, enabling us to jointly decode the embedded message and the original image signal perfectly. We also have performed extensive experiments to validate the superior embedding performance of our proposed RIDH method over encrypted domain.

REFERENCES

[1] Y. Q. Shi, X. Li, X. Zhang, H. Wu, and M. B., "Reversible data hiding: Advances in the past two decades," IEEE Access, vol. PP, no. 99, pp.1–1, 2016.
[2] Z. Hua, Y. Zhou, C.-M. Pun, and C. L. P. Chen, "2D Sine Logistic modulation map for image encryption," Information Sciences, vol. 297, no. 0, pp. 80–94, 2015.
[3] X. Zhang, "Reversible data hiding with optimal value transfer," IEEE Transactions on Multimedia, vol. 15, no. 2, pp. 316–325, 2013.
[4] J. Fridrich, M. Goljan, and R. Du, "Lossless data embedding: New paradigm in digital watermarking," EURASIP J. Appl. Signal Process., vol. 2002, no. 1, pp. 185–196, 2002.
[5] M. U. Celik, G. Sharma, A. M. Tekalp, and E. Saber, "Lossless generalized-LSB data embedding," IEEE Transactions on Image Processing, vol. 14, no. 2, pp. 253–266, 2005.
[6] Z. Ni, Y.-Q. Shi, N. Ansari, and W. Su, "Reversible data hiding," IEEE Transactions on Circuits and Systems for Video Technology, vol. 16, no. 3, pp. 354–362, 2006.
[7] A. M. Alattar, "Reversible watermark using the difference expansion of a generalized integer transform," IEEE Transactions on Image Processing, vol. 13, no. 8, pp. 1147–1156, 2004.
[8] Y. Hu, H.-K. Lee, and J. Li, "DE-based reversible data hiding with improved overflow location map," IEEE Transactions on Circuits and Systems for Video Technology, vol. 19, no. 2, pp. 250–260, 2009.
[9] J. Zhou, W. Sun, L. Dong, X. Liu, O. C. Au, and Y. Y. Tang, "Secure reversible image data hiding over encrypted domain via key modulation, "IEEE Transactions on Circuits and Systems for Video Technology, vol. 26, no. 3, pp. 441–452, 2016.
[10] Z. Qian, X. Zhang, and S. Wang, "Reversible data hiding in encrypted JPEG bitstream," IEEE Transactions on Multimedia, vol. 16, no. 5, pp. 1486–1491, 2014.
[11] K. Ma, W. Zhang, X. Zhao, N. Yu, and F. Li, "Reversible data hiding in encrypted images by reserving room before encryption," IEEE Transactions on Information Forensics and Security, vol. 8, no. 3, pp. 553–562, 2013.
[12] S. Yi and Y. Zhou, "Binary-block embedding for reversible data hiding in encrypted images," Signal Processing, vol. 133, pp. 40–51, 2017.

[13] X. Liao, K. Li, and J. Yin, "Separable data hiding in encrypted image based on compressive sensing and discrete fourier transform," Multimedia Tools and Applications, vol. 76, no. 20, pp. 20 739–20 753, 2017.
[14] X. Liao and C. Shu, "Reversible data hiding in encrypted images based on absolute mean difference of multiple neighboring pixels," Journal of Visual Communication and Image Representation, vol. 28, no. 0, pp. 21–27, 2015.
[15] M. Li, D. Xiao, Y. Zhang, and H. Nan, "Reversible data hiding in encrypted images using cross division and additive homomorphism, "Signal Processing: Image Communication, vol. 39, no. Part A, pp. 234–248, 2015.
[16] X. Zhang, "Reversible data hiding in encrypted image," IEEE Signal Processing Letters, vol. 18, no. 4, pp. 255–258, 2011.
[17] W. Zhang, K. Ma, and N. Yu, "Reversibility improved data hiding in encrypted images," Signal Processing, vol. 94, no. 0, pp. 118–127, 2014.
[18] X. Cao, L. Du, X. Wei, D. Meng, and X. Guo, "High capacity reversible data hiding in encrypted images by patch-level sparse representation," IEEE Transactions on Cybernetics, vol. 46, no. 5, pp. 1132–1143, 2016.
[19] S. Yi and Y. Zhou, "An improved reversible data hiding in encrypted images," 2015 IEEE China Summit and International Conference on Signal and Information Processing (ChinaSIP), pp. 225–229, 2015.
[20] T. Mathew and M. Wilscy, "Reversible data hiding in encrypted images by active block exchange and room reservation," in 2014 International Conference on Contemporary Computing and Informatics (IC3I), pp. 839–844.
[21] X. Zhang, "Separable reversible data hiding in encrypted image," IEEE Transactions on Information Forensics and Security, vol. 7, no. 2, pp. 826–832, 2012.
[22] Z. Yin, B. Luo, and W. Hong, "Separable and error-free reversible data hiding in encrypted image with high payload," The Scientific World Journal, vol. 2014, p. 8, 2014.
[23] Z. Yin, H. Wang, H. Zhao, B. Luo, and X. Zhang, "Complete separable reversible data hiding in encrypted image," Cloud Computing and Security: First International Conference, ICCCS 2015, pp. 101–110, 2015.
[24] F. Huang, J. Huang, and Y. Q. Shi, "New framework for reversible data hiding in encrypted domain," IEEE Transactions on Information Forensics and Security, vol. 11, no. 12, pp. 2777–2789, 2016.
[25] Z. Qian and X. Zhang, "Reversible data hiding in encrypted images with distributed source encoding," IEEE Transactions on Circuits and Systems for Video Technology, vol. 26, no. 4, pp. 636–646, 2016.
[26] W. Puech, M. Chaumont, and O. Strauss, "A reversible data hiding method for encrypted images," Security, Forensics, Steganography, and Watermarking of Multimedia Contents X, Proceedings of SPIE 6819, 2008.
[27] W. Hong, T.-S. Chen, and H.-Y. Wu, "An improved reversible data hiding in encrypted images using side match," IEEE Signal Processing Letters, vol. 19, no. 4, pp. 199–202, 2012.
[28] M. Li, D. Xiao, Z. Peng, and H. Nan, "A modified reversible data hiding in encrypted images using random diffusion and accurate prediction, "ETRI Journal, vol. 36, no. 2, pp. 325–328, 2014.
[29] X. Zhang, Z. Qian, G. Feng, and Y. Ren, "Efficient reversible data hiding in encrypted images," Journal of Visual Communication and Image Representation, vol. 25, no. 2, pp. 322–328, 2014.
[30] S. Zheng, D. Li, D. Hu, D. Ye, L. Wang, and J. Wang, "Lossless data hiding algorithm for encrypted images with high capacity," Multimedia Tools and Applications, pp. 1–14, 2015.
[31] X. Zhang, C. Qin, and G. Sun, "Reversible data hiding in encrypted images using pseudorandom sequence modulation," Digital Forensics and Watermaking, vol. 7809, pp. 358–367, 2013.
[32] X. Wu and W. Sun, "High-capacity reversible data hiding in encrypted images by prediction error," Signal Processing, vol. 104, pp. 387–400, 2014.
[33] Y.-C. Chen, C.-W. Shiu, and G. Horng, "Encrypted signal-based reversible data hiding with public key cryptosystem," Journal of Visual Communication and Image Representation, vol. 25, no. 5, pp. 1164 –1170, 2014.
[34] C.-W. Shiu, Y.-C. Chen, and W. Hong, "Encrypted image-based reversible data hiding with public key cryptography from difference expansion," Signal Processing: Image Communication, vol. 39, Part A, pp. 226–233, 2015.
[35] X. Zhang, J. Wang, Z. Wang, and H. Cheng, "Lossless and reversible data hiding in encrypted images with public key cryptography," IEEE Transactions on Circuits and Systems for Video Technology, vol. PP, no. 99, pp. 1–1, 2015.
[36] Z. Yin, A. Abel, J. Tang, X. Zhang, and B. Luo, "Reversible data hiding in encrypted images based on multi-level encryption and block histogram modification," Multimedia Tools and Applications, vol. 76, no. 3, pp.3899–3920, 2017.
[37] Z. Hua, S. Yi, and Y. Zhou, "Medical image encryption using high-speed scrambling and pixel adaptive diffusion," Signal Processing, vol. 144, pp. 134 – 144, 2018.
[38] Y. Zhou, L. Bao, and C. P. Chen, "A new 1D chaotic system for image encryption," Signal Processing, vol. 97, pp. 172 – 182, 2014.

BIOGRAPHIES

Marteena P.M.
D/O P.D Martin
DDMCA Student at
SNGIST N. Paravur, Ernalulam, Kerala, India