Research Journal of Information Technology

ISSN 1815-7432

Academic Journals Inc.

www.academicjournals.com

Research Journal of Information Technology 5 (2): 100-112, 2013
ISSN 1815-7432 / DOI: 10.3923/rjit.2013.100.112

Bio-hiding for Smart Swipe Card: A Secret Security

[1]Rengarajan Amirtharajan, [1]G. Aishwarya, [1]M. Sai Krishna Karthik, [2]V. Thanikaiselvan and [1]J.B.B. Rayappan
[1]School of Electrical and Electronics Engineering, SASTRA University, 613401, India
[2]School of Electronics Engineering, VIT University, Vellore-632014, Tamil Nadu, India

Corresponding Author: Rengarajan Amirtharajan, School of Electrical and Electronics Engineering, SASTRA University, 613401, India

ABSTRACT

To ensure secrecy in communication, especially if it is through image or any other digital file, attention should be paid on the tool concerned in transferring the file. It may entail intricate algorithms and cryptic measures. But the query here is how far it is multifarious? How thorny is it for the impostor to haul out the masked data? If these questions are genuinely answered by one, without doubt the algorithm is a rock. Here presented one such scheme which, of course, answers the above, ensures defence at three levels namely cryptography, steganography and fingerprinting. The first one involves encryption of text, steganography deals with burying the stealthy information, finally, fingerprinting, in general, toting up fingerprints to an entity or recognizing those which are previously inherent to a item. The efficacy is tested by the delineated image aspects. Additionally, this method can be put into action easily and bulky size of secret data can be passed on.

Key words: Authentication, data hiding, fingerprints, image steganography

INTRODUCTION

Authentication using biometric involves obtaining physical traits from an individual like finger marks, iris, facial, voice and retinal models, written monikers etc., for certifying an individual's identity via processing images. Important here is mentioning that characteristics obtained are unique to an individual and remain so in his life time. Finger-print based biometric authentication system has in this regard evolved as one of the oldest and most mature system due to the unique and reliable finger-print of a person and successfully implemented in many varied applications (Cheddad *et al*., 2008). The fingerprints analysis in favour of harmonizing intention usually involves likening of quite a few traits in the feature prototype. They comprise models, having amassed features like crumples (raised skin), furrows (lowered skin) and minutia points, which are distinctive traits that create inside such moulds. Indeed it becomes crucial of having knowledge on construction plus assets belonging to individual with the intention of effectively utilizing a number of technologies in imaging. First and foremost Minutia characteristics in finger mark creases: ridge ending, bifurcation, along with short ridge (or dot). Bifurcation is nothing but the point where a solitary ridge segments to two crinkles. Details and trivia remain incredibly imperative to examine fingerprint because no two finger marks are alike.

Two factors come into play for deciding the suitably of finger-print based biometrics than others. These are space complexity and time complexity (Brindha and Vennila, 2011). A bare minimum

sized fingerprint stencil has size of bytes more than a few hundred where customary smart-cards possess nonvolatilizable memory of 8K up to 16K approximately. Hence, place intricacy is not a chief crisis for as latter is able to hoard whole finger mark templet. In case of period complication, its processor should be gifted of accomplishing an intact fingerprint toning algorithm instantaneously.

The main aim of this paper is to hide the fingerprint in the person's photo which can be later printed on his/her Smart card or ID card for identification as well as authentication purpose. In addition, this work mainly focused on the software implementation of hiding the fingerprint in the image for authentication purpose (Yang *et al*., 2007). To enhance security, prior to embedding, the fingerprint is encrypted (Salem *et al*., 2011) and then embedded (Amirtharajan and Rayappan, 2012a; b ; c ; d). The fingerprint will be encrypted using an encryption algorithm which will be later discussed. Then the encrypted bits get rooted in covers through Scattered (LSB) embedding. Rather than infixing covert bits in linear fashion as in casual LSB substitution method (Chan and Cheng, 2004), here it is done so in a unique non-linear fashion based on a PRNG which gets generated according to covers' size.

Cryptography (Schneier, 2007), Steganography (Stefan and Fabin, 2000; Thenmozhi *et al*., 2012; Zhu *et al*., 2011; Zhao and Luo, 2012) and Biometrics are blend together to implement a smart Identity Card (ID card) by hiding the fingerprint in the person's photo which can be later printed on his/her Smart card or ID card for identification as well as authentication purpose. Thus the stored/hidden biometric template (finger print) can be compared with the live template for authentication purpose and the photo is an indication for his/her identity. Steganography (Cheddad *et al*., 2010), watermarking (Abdulfetah *et al*., 2010; Zeki *et al*., 2011) and Cryptography (Salem *et al*., 2011; Zaidan *et al*., 2010) play a vital role in providing security (Hmood *et al*., 2010a, b; Rajagopalan *et al*., 2012).

Steganography can be broadly classified into spatial (Luo *et al*., 2011; Mohammad *et al*., 2011) and frequency domain (Amirtharajan and Rayappan, 2012d; Provos and Honeyman, 2003; Thanikaiselvan *et al*., 2011a). Another classification is based on the cover object (Bender *et al*., 1996) hike text (Al-Azawi and Fadhil, 2010; Xiang *et al*., 2011), video (Al-Frajat *et al*., 2010), audio (Zhu *et al*., 2011) or in an image (Gutub, 2010; Amirtharajan *et al*., 2012; Janakiraman *et al*., 2012a, b; Padmaa *et al*., 2011; Zaidan *et al*., 2010; Zanganeh and Ibrahim, 2011; Thanikaiselvan *et al*., 2011b). In this study, an optimistic effort has been taken to encrypt the finger print through ingenious symmetric key crypto system, later embedded in the photo to build smart ID cards.

Some useful cryptographic system: Since a Simple XOR Cipher with a single key can easily be guessed by the intruders, we have implemented a three key Cryptographic algorithm. The Encryption and Decryption algorithms will be discussed in this section.

Encryption algorithm: Figure 1 shows the encryption algorithm with 3 keys. It consists of 3 different keys say 1, 2 and 3. The first one will be the user defined key i.e., the user has to enter the key he wants to use for encrypting the fingerprint image (INPUT). The other two keys key 2 and key 3 will be the 1-bit left rotated versions of the previous ones. That means key 2 will be the 1-bit left rotated version of key 3 and soon as shown in the above figure.

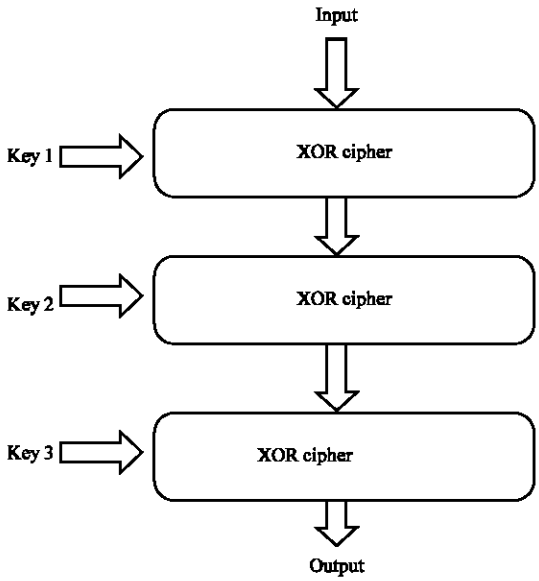

Fig. 1: Encryption algorithm with 3 different keys

As per the Figure, the fingerprint image will be first XORed with key 1 and the output of the first XOR Cipher will be again XORed with key 2 and soon. The resulting output will be the encrypted image which was XORed three times using a set of 3 different keys. This is about the encryption algorithm used for encrypting the fingerprint image.

Algorithm: Fingerprint encrypting procedure

Step 1: Read the fingerprint image which is going to be embedded
Step 2: Convert the coloured fingerprint image into gray image
Step 3: Now convert the converted image into a binary stream
Step 4: Input the key 1 for encrypting the read fingerprint image
Step 5: Rotate key 1 to left (1-bit rotation) to get key 2
Step 6: Rotate key 2 to left (1-bit rotation) to get key 3
Step 7: Convert all the obtained keys into binary stream
Step 8: The converted finger print image binary stream is XORed with KEY 1 binary stream
Step 9: Repeat step 8 with keys key 2 and key 3 as shown in Fig. 2
Step 10: The resulting output will be the encrypted fingerprint image

Decryption algorithm: The decryption algorithm will be the same as that of the encryption algorithm which we have discussed earlier but in the reverse order as shown in the Fig. 1. The encrypted fingerprint image will be the input to the decryption algorithm and the keys are given in the reverse order.

The same procedure will be repeated for the decryption algorithm but the only change is that here the input to the first XOR cipher will be the encrypted fingerprint image and the key to it will be key 3. The same procedure is repeated as shown in Fig. 1 and the resulting output will be the original fingerprint image.

Algorithm: Fingerprint decrypting procedure

Step 1: Read the encrypted fingerprint output
Step 2: Now convert encrypted output to a binary stream
Step 3: Input the key 1 used for encrypting the original fingerprint image
Step 4: Rotate key 1 to left (1-bit rotation) to get key 2
Step 5: Rotate key 2 to left (1-bit rotation) to get key 3
Step 6: Convert all the obtained keys into binary stream
Step 7: The encrypted finger print image binary stream is XORed with key 3 binary stream
Step 8: Repeat Step 8 with keys KEY 2 and KEY 1 as shown in Fig. 2
Step 9: The resulting output will be the original fingerprint image

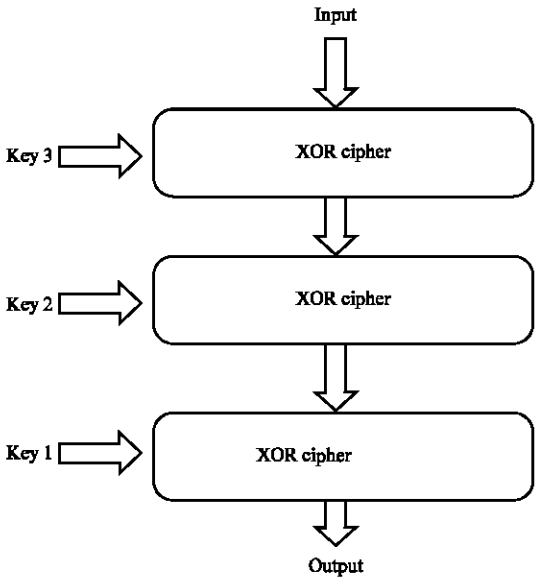

Fig. 2: Decryption algorithm with 3 different keys

PROPOSED METHOD

LSB is a simplest plus effective technique in data embedding technique. It directly embeds the secret bits in LSB of cover image. Even it introduces distortion when the embedded bits are more than three. This classical paradigm buries the data within cover via., engaging the bit stream of the message to substitute that of the cover's LSB sequentially. Image steganography intends for preserving numerical traits of congregating image in order to withstand or foil steganalysis. Nevertheless, LSB routines pioneer a little deformation in geometric properties of carrier signal or image to vindicate manoeuvring by means of steganalysis runs. To facilitate this susceptibility, a modus operandi as shown in Fig. 3 which performs scattered LSB embedding in addition to preservation of cover images' histogram is proposed.

In this technique, the secret is going to be embedded in a non-linear fashion based on Pseudo Random Sequence Generation (PRNG) contrary to the classical LSB steganography, where the confidential data is embedded inside a carrier file in a linear fashion. The pseudo random number is generated depending upon the size of the carrier. This process will be same as that of the classical LSB substitution method but the secret will be embedded in the pixels as decided by the pseudo random number. This pseudo random number must be produced during the extraction process.

As discussed earlier, the main aim of this study is to hide the fingerprint of a person in his/her own photo. For embedding the fingerprint in the cover image, a new Steganographic technique

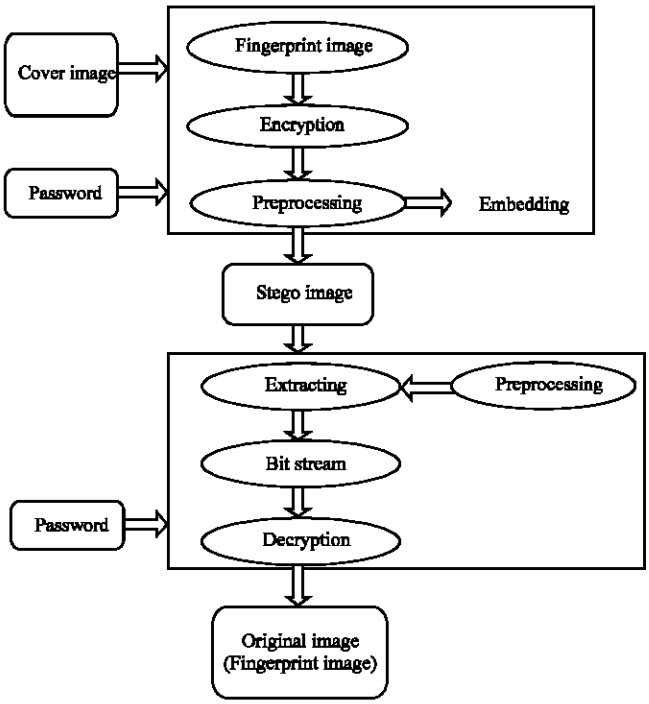

Fig. 3: Block diagram of proposed method

Scattered LSB embedding has been proposed. This model suggests a method, where the surreptitious information gets rooted in the non-linear fashion as discussed above. As an improvement, prior to embedding, the bits are encrypted and then embedded.

ENCRYPTION AND DATA HIDING

Algorithm: Embedding process

Step 1:	Study the finger print input as shown in Fig. 4
Step 2:	Get key from user for encryption purpose
Step 3:	Encrypt the fingerprint image with the key specified by the user and the encryption algorithm which was discussed earlier
Step 4:	Read the cover image into which the fingerprint has to be embedded
Step 5:	Generate a Pseudo random number as per the cover image's size
Step 6:	Get the number of bits for embedding in host image
Step 7:	Embed the encrypted fingerprint image into the cover image using scattered LSB technique as specified earlier
Step 8:	The resulting image will be the Stego image containing the fingerprint image in it

Algorithm: Extracting/retrieving process

Step 1:	Study the Stego output
Step 2:	Generate pseudo random number as per cover's size to know where fingerprint is hidden as shown in Fig. 5
Step 3:	Get the no: of bits embedded per pixel of cover image from the user
Step 4:	Retrieve the finger print image from the cover image using the same method used for embedding it in the cover image
Step 5:	Get the key from the user to decrypt the fingerprint image
Step 6:	Decrypt the fingerprint image using the key and the decryption algorithm as specified earlier
Step 7:	The resultant is the needed Fingerprint Image

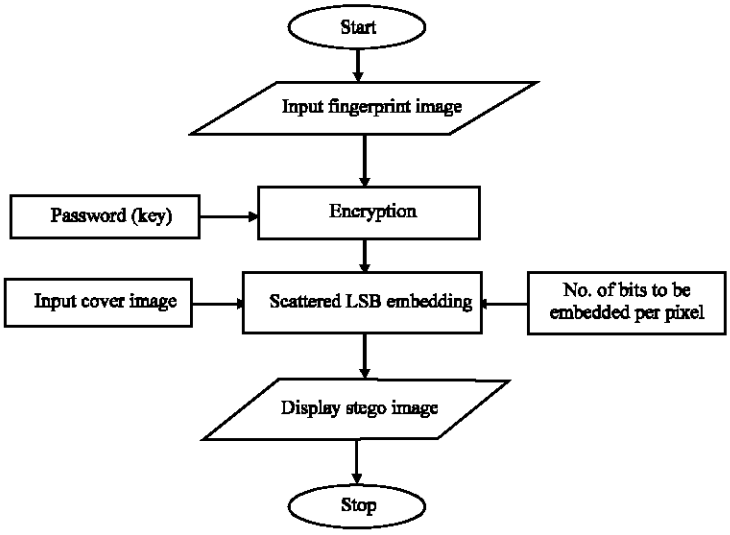

Fig. 4: Flowchart illustrating the embedding process along with encryption block

Obtained fingerprint output from retrieval process and the original fingerprint can be compared later:

- Flow charts
- Embedding process
- Extracting/retrieving process

The final result will be the stego image containing the encrypted fingerprint image in the cover image. This stego image can later be printed on ID cards or Smart card for authentication as well as identification purpose. The encrypted fingerprint image is retrieved back using this defined extracting process and decrypted using decryption algorithm. Then the decrypted fingerprint image can be compared with the original fingerprint image.

RESULTS AND DISCUSSION

To execute the algorithm four different fingerprints are taken which are of unique dimensions as shown in Fig. 6. These four are encrypted using four keys. The encryption keys are defined as 68, 89, 142, 222. The encrypted fingerprints are represented in Fig. 7 which w ill be embedded in the covers.

Four 256×256 gray images are chosen as cover as shown in Fig. 8. The embedded results (stego images) are shown for k = 1, 2, 3, 4 bit embedding are given in Fig. 9, 10, 11 and 12, respectively. One cannot visualize the artifacts as the stego result is very much closer to the cover.

At the same time, if we see the recovery part, though it is not a cake walk, we have presented the extracted results and finally decrypted fingerprints. One can go for more complex procedures that may include other faces of mathematics, encryption, steganography and advanced computing:

- For K = 1 bit embedding
- For K = 2 bit embedding
- For K = 3 bit embedding
- For K = 4 bit embedding

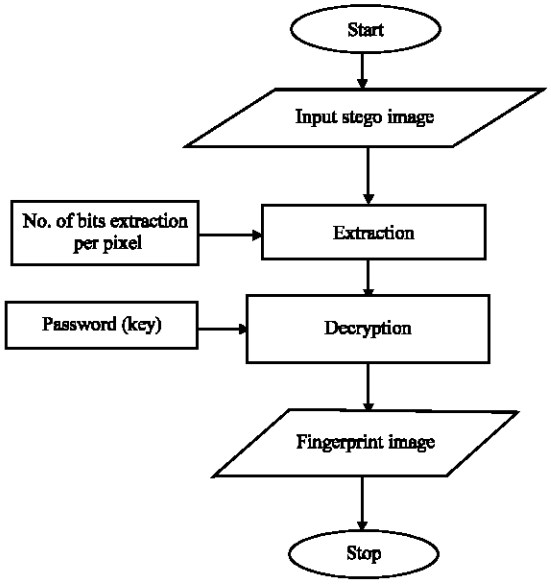

Fig. 5: Flowchart illustrating the extracting process along with the decryption block

Table 1: MSE and PSNR values of random embedding

Cover Image	No. of bits embedding (K)	MSE	PSNR
Student 1	K = 1	0.3336	52.8991
	K = 2	0.8278	48.9514
	K = 3	2.2627	44.5845
	K = 4	6.7049	39.8669
Student 2	K = 1	0.3703	52.4453
	K = 2	1.0145	48.0682
	K = 3	2.2627	44.5845
	K = 4	6.7049	39.8669
Student 3	K = 1	0.3703	52.4453
	K = 2	1.0145	48.0682
	K = 3	2.8411	43.5960
	K = 4	8.6805	38.7454
Student 4	K = 1	0.3790	52.3439
	K = 2	0.8433	48.8709
	K = 3	2.6848	43.8417
	K = 4	6.9953	39.6827

MSE and PSNR values for each image: The tabulated result confirms that, if amount of bits in embedding gets increased, Mean Square Error is increasing resulting in decreasing PSNR values as in Table 1. If we compare the MSE and PSNR values of the four taken covers, it is observed that PSNR for student 1 is relatively high.

Indeed it points out that, of all the four, the first image has high imperceptibility. It escapes human perception and free from visual artifacts. Thus stego and original cover remain the same thus making it a tricky task for the invader.

Extracted images from the stego images: Recovery process is done through by the same keys used for encryption and the results are displayed in Fig. 13 and 14.

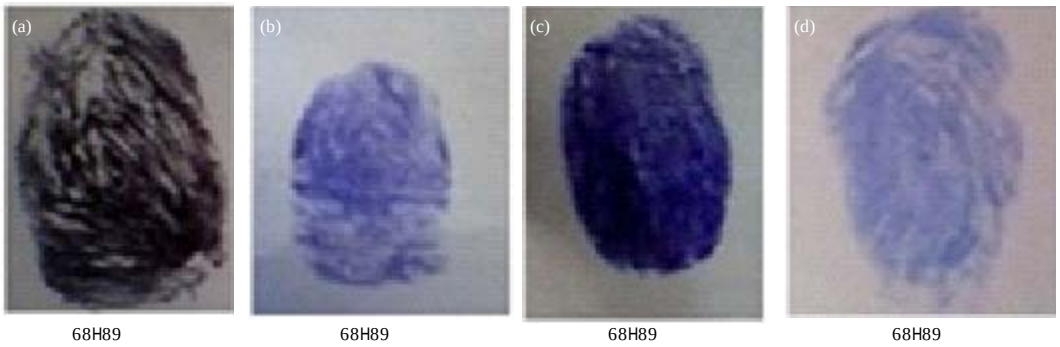

Fig. 6(a-d): Fingerprint images (a) Student 1, (b) Student 2, (c) Student 3 and (d) Student 4

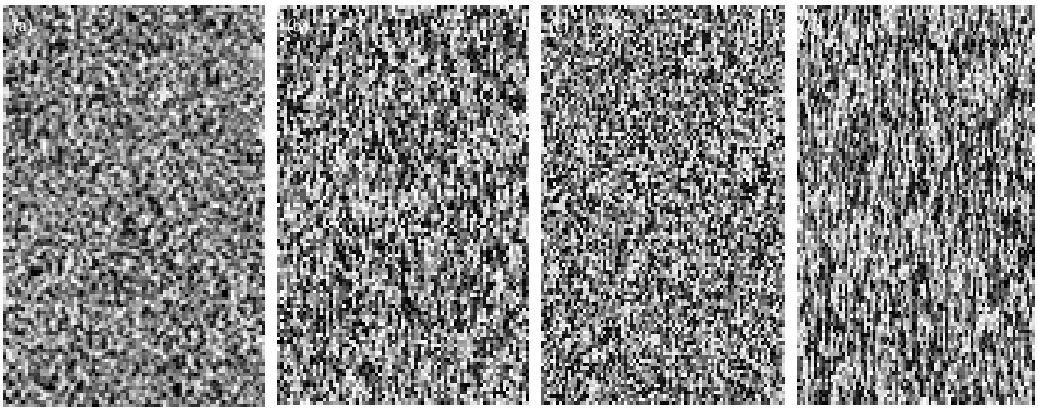

Fig. 7(a-d): Encrypted fingerprint images (a) Student 1, (b) Student 2, (c) Student 3 and (d) Student 4

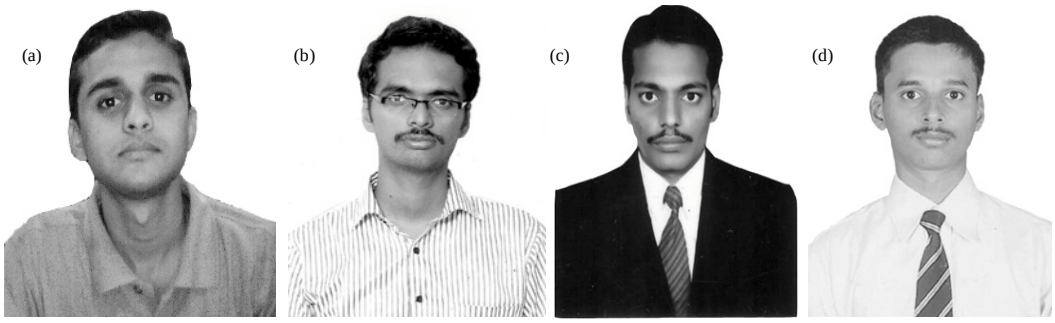

Fig. 8(a-d): Cover images (a) Student 1, (b) Student 2, (c) Student 3 and (d) Student 4

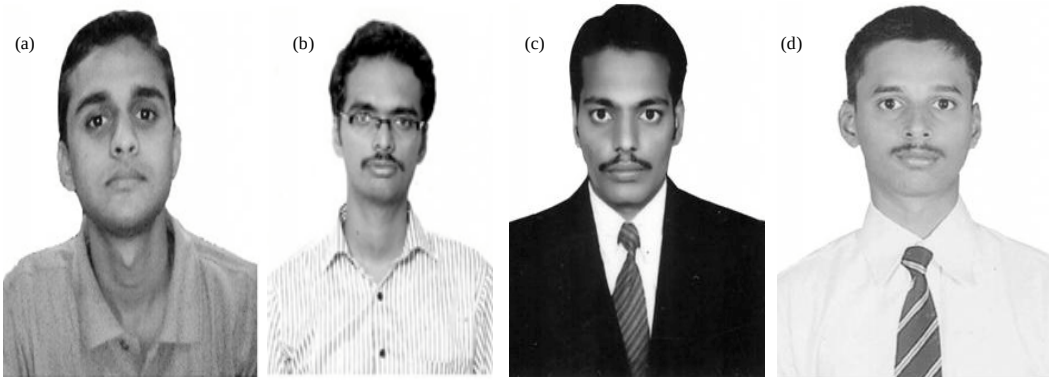

Fig. 9(a-d): Stego images for K = 1 bit, (a) Student 1, (b) Student 2, (c) Student 3 and (d) Student 4

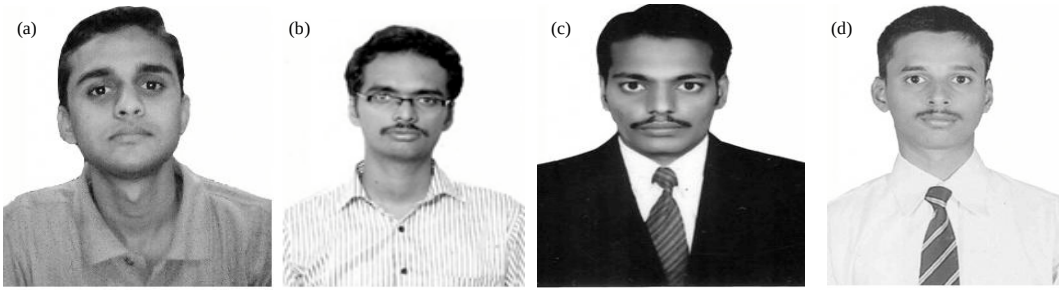

Fig. 10(a-d): Stego images for K = 2 bit, (a) Student 1, (b) Student 2, (c) Student 3 and (d) Student 4

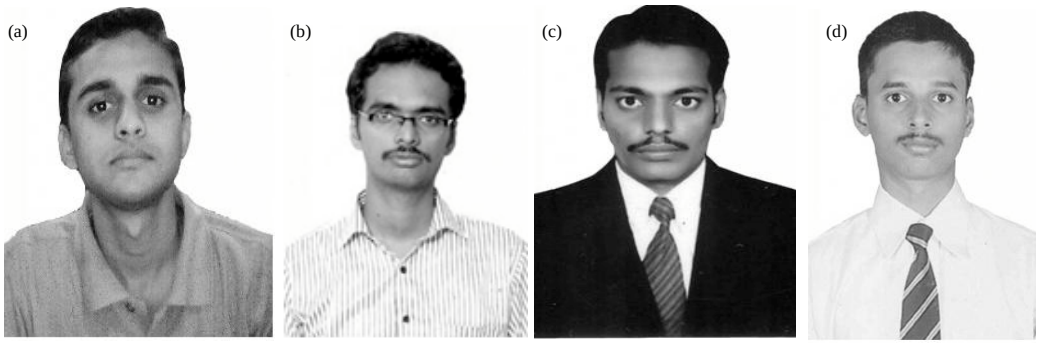

Fig. 11(a-d): Stego images for K = 3 bit, (a) Student 1, (b) Student 2, (c) Student 3 and (d) Student 4

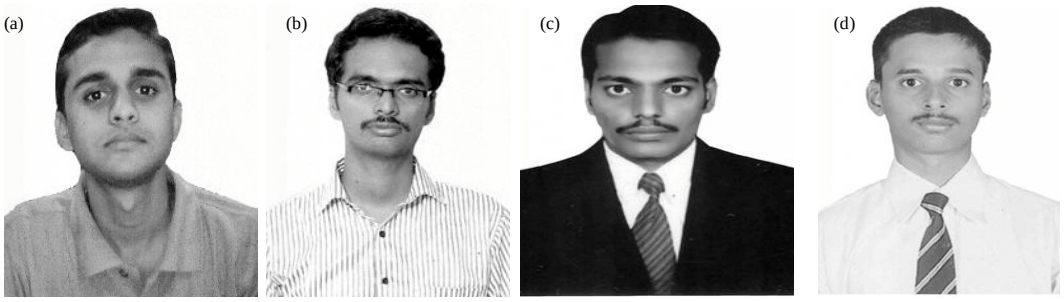

Fig. 12(a-d): Stego images for K = 4 bit, (a) Student 1, (b) Student 2, (c) Student 3 and (d) Student 4

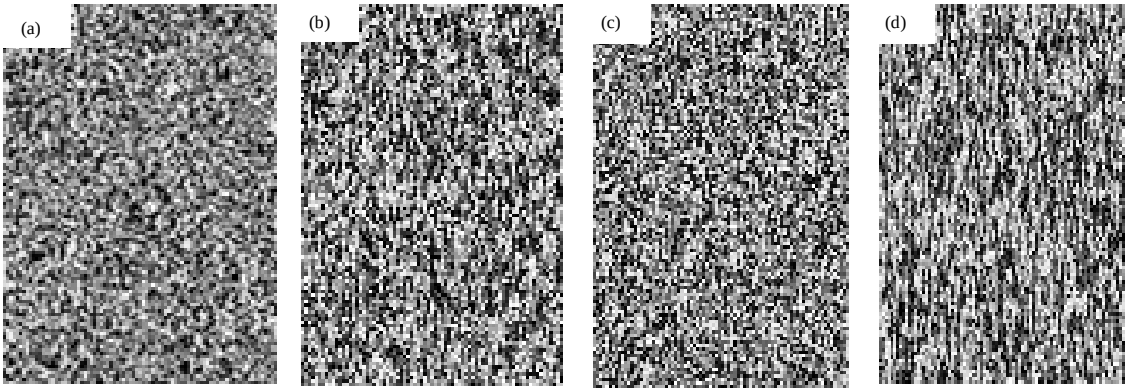

Fig. 13(a-d): Extracted images from the stego images (a) Student 1, (b) Student 2, (c) Student 3 and (d) Student 4

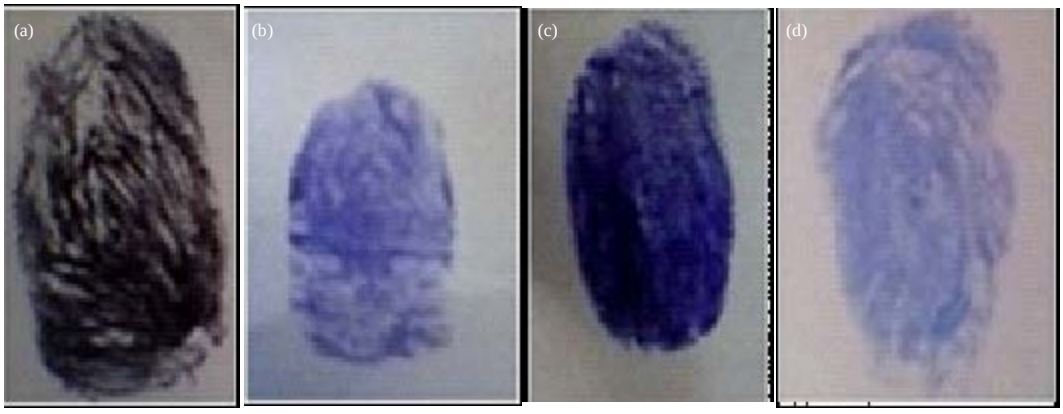

Fig. 14(a-d): Decrypted fingerprint images after extraction (a) Student 1, (b) Student 2, (c) Student 3 and (d) Student 4

The fingerprint images of four students are taken and are encrypted using the encryption algorithm as specified earlier with the keys as the last three digits for the register numbers. The encrypted fingerprint images are embedded in their pictures using the scattered LSB Steganographic Technique. As opposed to rooting secret bits in hosting image in linear fashion as in casual LSB substitution method, this methodology embeds encrypted data bits within the cover in some non-linear fashion based on a pseudo random sequence generation. It is so generated according to cover files' range; lying upon this PRNG, bits for embedding inside pixels of cover file is decided.

The final output will be result will be the stego image containing the encrypted fingerprint image in the cover image. This stego image can later be printed on ID cards or Smart card for authentication as well as identification purpose. The encrypted fingerprint image gets retrieved back as of the stego output image using extracting formula and decrypted using decryption algorithm. Then the decrypted fingerprint image can be compared with the original fingerprint image.

CONCLUSION

In this study, the available Cryptographic and Steganographic techniques to hide the fingerprint (Brindha and Vennila, 2011; Chan and Cheng, 2004) of a person in his/her own image and this can later be printed on ID cards or Smart card for authentication as well as identification purpose. From the printed stego image, the fingerprint can be retrieved back and can be compared with the hive fingerprint. Thus the two techniques namely encryption and PRNG based embedding are used in LSB embedding to enhance its security (Yang *et al*., 2007; Hmood *et al*., 2010b). The experimental values of MSE and PSNR for each case are given in the results.

REFERENCES

Abdulfetah, A.A., X. Sun, H. Yang and N. Mohammad, 2010. Robust adaptive image watermarking using visual models in DWT and DCT domain. Inform. Technol. J., 9: 460-466.

Al-Azawi, A.F. and M.A. Fadhil, 2010. Arabic text steganography using kashida extensions with huffman code. J. Applied Sci., 10: 436-439.

Al-Frajat, A.K., H.A. Jalab, Z.M. Kasirun, A.A. Zaidan and B.B. Zaidan, 2010. Hiding data in video file: An overview. J. Applied Sci., 10: 1644-1649.

Amirtharajan, R. and J.B.B. Rayappan, 2012a. An intelligent chaotic embedding approach to enhance stego-image quality. Inform. Sci., 193: 115-124.

Amirtharajan, R. and J.B.B. Rayappan, 2012b. Brownian motion of binary and gray-binary and gray bits in image for stego. J. Applied Sci., 12: 428-439.

Amirtharajan, R. and J.B.B. Rayappan, 2012c. Inverted pattern in inverted time domain for icon steganography. Inform. Technol. J., 11: 587-595.

Amirtharajan, R. and J.B.B. Rayappan, 2012d. Pixel authorized by pixel to trace with SFC on image to sabotage data mugger: A comparative study on PI stego. Res. J. Inform. Technol., 4:: 124-139.

Amirtharajan, R., J. Qin and J.B.B. Rayappan, 2012. Random image steganography and steganalysis: Present status and future directions. Inform. Technol. J., 11: 566-576.

Bender, W., D. Gruhl, N. Morimoto and A. Lu, 1996. Techniques for data hiding. IBM Syst. J., 35: 313-336.

Brindha, S. and I. Vennila, 2011. Hiding fingerprint in face using scattered LSB embedding steganographic technique for smart card based authentication system. Int. J. Comput. Appl., 26: 51-55.

Chan, C.K. and L.M. Cheng, 2004. Hiding data in images by simple LSB substitution. J. Pattern Recognit. Soc., 37: 469-474.

Cheddad, A., J. Condell, K. Curran and P. McKevitt, 2008. Biometric inspired digital image steganography. Proceedings of the 15th Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems, March 31-April 4, 2008, Belfast, Northern Ireland, pp: 159-168.

Cheddad, A., J. Condell, K. Curran and P.M. Kevitt, 2010. Digital image steganography: Survey and analysis of current methods. Signal Process., 90: 727-752.

Gutub, A.A.A., 2010. Pixel indicator technique for RGB image steganography. J. Emerging Technol. Web Intell., 2: 56-64.

Hmood, A.K., B.B. Zaidan, A.A. Zaidan and H.A. Jalab, 2010a. An overview on hiding information technique in images. J. Applied Sci., 10: 2094-2100.

Hmood, A.K., H.A. Jalab, Z.M. Kasirun, B.B. Zaidan and A.A. Zaidan, 2010b. On the Capacity and security of steganography approaches: An overview. J. Applied Sci., 10: 1825-1833.

Janakiraman, S., R. Amirtharajan, K. Thenmozhi and J.B.B. Rayappan, 2012a. Firmware for data security: A review. Res. J. Inform. Technol., 4: 61-72.

Janakiraman, S., R. Amirtharajan, K. Thenmozhi and J.B.B. Rayappan, 2012b. Pixel forefinger for gray in color: A layer by layer stego. Inform. Technol. J., 11: 9-19.

Luo, H., Z. Zhao and Z.M. Lu, 2011. Joint secret sharing and data hiding for block truncation coding compressed image transmission. Inform. Technol. J., 10: 681-685.

Mohammad, N., X. Sun and H. Yang, 2011. An excellent Image data hiding algorithm based on BTC. Inform. Technol. J., 10: 1415-1420.

Padmaa, M., Y. Venkataramani and R. Amirtharajan, 2011. Stego on 2^n: 1 Platform for users and embedding. Inform. Technol. J., 10: 1896-1907.

Provos, N. and P. Honeyman, 2003. Hide and seek: An introduction to steganography. IEEE Secur. Privacy, 1: 32-44.

Rajagopalan, S., R. Amirtharajan, H.N. Upadhyay and J.B.B. Rayappan, 2012. Survey and analysis of hardware cryptographic and steganographic systems on FPGA. J. Applied Sci., 12: 201-210.

Salem, Y., M. Abomhara, O.O. Khalifa, A.A. Zaidan and B.B. Zaidan, 2011. A review on multimedia communications cryptography. Res. J. Inform. Technol., 3: 146-152.

Schneier, B., 2007. Applied Cryptography: Protocols, Algorithm and Source Code in C. 2nd Edn., Wiley, India.

Stefan, K. and A. Fabin, 2000. Information Hiding Techniques for Steganography and Digital Watermarking. Artech House, London, UK.

Thanikaiselvan, V., P. Arulmozhivarman, R. Amirtharajan and J.B.B. Rayappan, 2011a. Wave (let) decide choosy pixel embedding for stego. Proceedings of the International Conference on Computer, Communication and Electrical Technology, March 18-19, 2011, India, pp: 157-162.

Thanikaiselvan, V., S. Kumar, N. Neelima and R. Amirtharajan, 2011b. Data battle on the digital field between horse cavalry and interlopers. J. Theor. Applied Inform. Technol., 29: 85-91.

Thenmozhi, K., P. Praveenkumar, R. Amirtharajan, V. Prithiviraj, R. Varadarajan and J.B.B. Rayappan, 2012. OFDM+CDMA+Stego = Secure Communication: A Review. Res. J. Inform. Technol., 4: 31-46.

Xiang, L., X. Sun, Y. Liu and H. Yang, 2011. A secure steganographic method via multiple choice questions. Inform. Technol. J., 10: 992-1000.

Yang, C.N., T.S. Chen, K.H. Yu and C.C. Wang, 2007. Improvements of image sharing with steganography and authentication. J. Syst. Software, 80: 1070-1076.

Zaidan, B.B., A.A. Zaidan, A.K. Al-Frajat and H.A. Jalab, 2010. On the differences between hiding information and cryptography techniques: An overview. J. Applied Sci., 10: 1650-1655.

Zanganeh, O. and S. Ibrahim, 2011. Adaptive image steganography based on optimal embedding and robust against chi-square attack. Inform. Technol. J., 10: 1285-1294.

Zeki, A.M., A.A. Manaf and S.S. Mahmod, 2011. High watermarking capacity based on spatial domain technique. Inform. Technol. J., 10: 1367-1373.

Zhao, Z. and H. Luo, 2012. Reversible data hiding based on Hilbert curve scan and histogram modification. Inform. Technol. J., 11: 209-216.

Zhu, J., R.D. Wang, J. Li and D.Q. Yan, 2011. A huffman coding section-based steganography for AAC audio. Inform. Technol. J., 10: 1983-1988.