A Secure High-Capacity Video Steganography Using Bit Plane Slicing Through (7, 4) Hamming Code

Ananya Banerjee and Biswapati Jana

Abstract Achievement of high-capacity data hiding using a digital media is an important research issue in the field of steganography. In this paper, we have introduced a novel scheme of data hiding directly within the video stream **using** bit plane slicing through (7, 4) Hamming code with the help of shared secret key. In the proposed scheme, a secret logo image is embedded within the cover video stream for authentication and ownership identification through Hamming code based video steganography. Each frame of secret video has been separated into individual three basic color blocks (R, G and B) and then partitioned into (3 × 3) pixel blocks. After that, each color block is sliced up into 4 bit planes starting from LSB plane. The pixels' positions of cover images are randomly selected by Pseudorandom Number Generator (PRNG) using a shared secret seed value and data embedding performed using (7, 4) Hamming code. As a result, 36 bits secret data can be embedded within a (3 × 3) pixel block which is almost eight times greater than Ramadhan and Khaled's scheme (Systems, applications and technology conference (LISAT), 2014 IEEE Long Island, 2014) [1]. Here, we achieve a high payload with good visual quality stego video. **Furthermore, the video compression is lossless so the video file size is strictly preserved for post-data embedding**.

Keywords Video steganography · Hamming code · Least significant bit (LSB) Bit plane · Data hiding

A. Banerjee (✉) · B. Jana
Department of Computer Science, Vidyasagar University,
Midnapore 721102, West Bengal, India
e-mail: anaanya.2011@gmail.com

B. Jana
e-mail: biswapatijana@gmail.com

S. Bhattacharyya et al. (eds.), *Advanced Computational and Communication Paradigms*, Advances in Intelligent Systems and Computing 706,
https://doi.org/10.1007/978-981-10-8237-5_9

1 Introduction

Steganography is the art and science of hidden data communication. Till date, many data hiding schemes [2] are developed but only few of them are considered being more secured and have less distortion. The data hiding schemes are useful in many application areas to solve the problem of ownership identification, copyright protection, authentication, verification, and more. The main aims of data hiding schemes are to ensure extraction of secret data and recovery of original object from stego media. On the other hand, data should stay hidden in stego media even if the eavesdropper tampered the stego or degrading through natural phenomenon like transmission resampling, compression, or filtering, etc. The main drawbacks of data hiding schemes are not to provide a good solution in such cases. The degree of distortion will be high due to increase of data embedding capacity that should be balanced mathematically using spread spectrum. The data embedding in video is considered to be more unsuspicious and secured and less exploration has been done till today in this research area using Hamming code.

2 Related Work

Video hiding inside a video stream using nonuniform rectangular partition is done by Sheng et al. [3]. Then, another video hiding scheme is proposed by Yadav et al. [4] based on LSB technique which replaces the least significant bits of pixels selected to hide the secret information. Video as a collection of numerous frames has greater data hiding capacity as the small color change in the whole video stream is hard to detect in human eyes. Dasgupta et al. [5] proposed hash-based LSB techniques in spatial domain where the bits of the message can be inserted in intensity pixels of the video in LSB positions. Here, we have proposed video steganography using (7, 4) Hamming code for color images. We have divided R, G, B color pixels in bit plane [6] starting from LSB to LSB-3 (up to 4 bit plane) partitioned into (3 × 3) blocks and then apply Hamming code based data hiding scheme. In this scheme, 36 bits of data are embedded within nine pixels which is more higher than other existing LSB technique [1, 4, 5] and most of the LSB techniques are prone to attack [7, 8]. Also, it maintains high visual quality. Additionally to provide more security, the message is encrypted using symmetric key encryption techniques. Thus, we have achieved secure steganographic system for hiding data in video stream using both cryptography and steganography techniques.

3 Proposed Method

A video stream consists of collection of frames and the secret data is embedded in these frames as payload. The cover video is broken down into frames before embedding. Each frame is now considered as a cover image. Now, the proposed technique has been applied to conceal the secret data in the carrier frames.

Suppose, I is considered as the cover frame image of size (M × N), and I' is the marked image after embedding data $D = \{d_1, \ldots, d_X\}$, where $d_i \in \{0, 1\}$, $1 \leq i \leq X$. Here, H is a parity check matrix of the Hamming code. Let H be

$H = \begin{vmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{vmatrix}$, Embedding capacity is an important metric for data embedding. It is measured by the number of secret bits that can be embedded into a cover image. The embedding capacity is calculated as [2] $ER = L/M \times N$ bpp where L is the length of the secret message. Before embedding the secret data, we take 36 bit secret key k_1 which is known to both the sender and the receiver, to encrypt the secret data bit using symmetric key encryption. We have taken each pixel block of size (3 × 3) and 4 bit plane of each pixel is used to embed the data, which results in (3 × 3) × 4 = 36 bits of data (D_1) in one iteration. As an additional security measure, instead of choosing the cover image pixel block serially, we will use Pseudorandom Number Generator (PRNG) function with a secret predefined seed k_2 (which is only known to the sender and the receiver) to determine the next available block for embedding. Since this seed will be known to the sender and receiver only, the generated unique pattern of pixel block selection can be used in embedding and extraction process securely. The data embedding procedure is enlisted in Algorithm 1 and the data extraction procedure is depicted in Algorithm 2 (Fig. 1).

Algorithm 1: Data embedding process

Input: Cover video, secret data bits D, Hamming matrix H, secret key k_1, and seed value k_2

Output: A stego video

Step 1: Extract each frame from video stream as a color image I of size (M × N).
Step 2: Collect random sequence of pixel blocks of size (3 × 3) from cover image $I_{M \times N}$ using PRNG (k_2). Say the pixel blocks are $X_1, X_2, \ldots, X_{MN}$.
Step 3: Convert X_i into three separate RGB color blocks X_{iR}, X_{iG}, and X_{iB}.
Step 4: Convert each X_i's into binary form.
Step 5: Perform bit plane slicing of each X_i 's up to 4 bit plane starting from LSB that is $X_{iR(LSB),}$ $X_{iR(LSB-1),}$ $X_{iR(LSB-2),}$ $X_{iR(LSB-3)}$.
Step 6: Take $c = X_{iR(LSB)}$ and calculate the syndrome $S_1 = (H \times (c)^{T)T}$.
Step 7: Perform $D_1' = (D_1 \oplus k_1)$; k_1 = 36 bit length and D_1 is also same length.

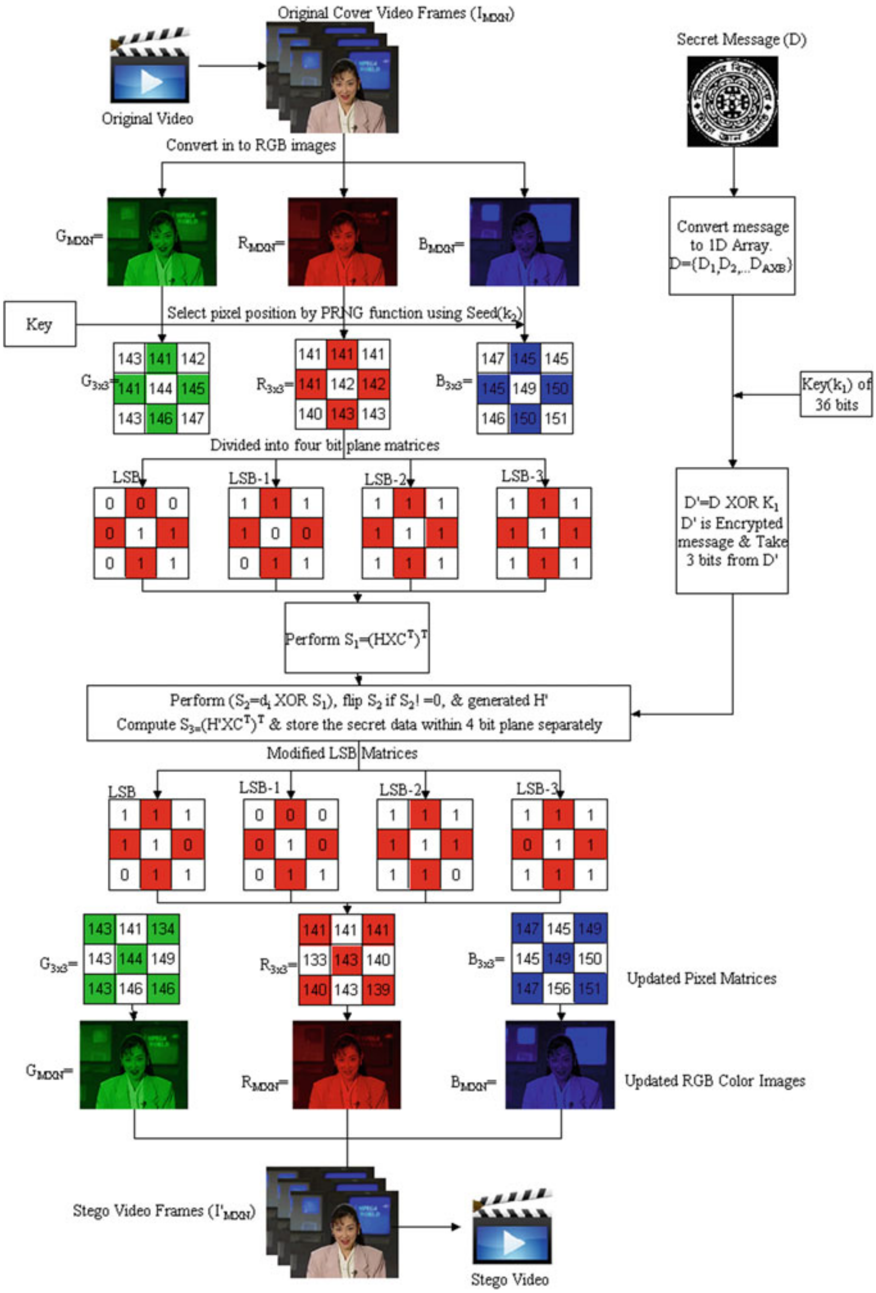

Fig. 1 Pictorial diagram of the proposed data hiding scheme

```
Step 8:  Take 3 bits secret data $d_i = \{d_1, d_2, d_3\}$ from $D_1$’ where $d_i \in \{0, 1\}$.
Step 9:  Calculate $S_2 = (d_i \oplus S_1)$; if $S_2 = 0$, no change, otherwise flip a bit at the positional value of $S_2$ and generate H’.
Step 10: Compute $S_3 = (H' \oplus c)$ and store the data.
Step 11: Replace the matrix (c) with $S_3$ and update $X_{iR(LSB)}$.
Step 12: Repeat Step 4 to 10 using $X_{iR(LSB-1)}$, $X_{iR(LSB-2)}$ and $X_{iR(LSB-3)}$.
Step 13: Repeat Step 5 to 11 to embed secret data on $X_{iG}$ and $X_{iB}$ color blocks.
Step 14: Repeat Step 2 to 12 to embed secret data on each and every random sequence of ($X_i$’s) of pixel blocks.
Step 15: Finally, after combining each stego block, we get stego frame (I’) of size (M × N).
Step 16: Generate stego video stream with encoded frames.
Step 17: End.
```

Algorithm 2: Data extraction process

Input: Stego video stream, Hamming matrix H, secret key k_1, and seed value k_2
Output: Original Secret Message D.

```
Step 1:  Convert the video stream into frames. Extract each frame as a color cover image I’ of size (M × N).
Step 2:  Use PRNG with predetermined seed $k_2$ to determine the stego pixel of random sequence $X'_i$ of size [3 × 3] from stego image I’.
Step 3:  Separate RGB components into $X'_{iR}$, $X'_{iG}$, $X'_{iB}$.
Step 4:  Convert into binary form of each $X'_{iR}$, $X'_{iG}$, and $X'_{iB}$.
Step 5:  Perform 4 bit plane slicing of each $X'_i$‘s starting from LSB that is $X'_{iR\,(LSB)}$, $X'_{iR(LSB-1)}$, $X'_{iR(LSB-2)}$, $X'_{iR(LSB-3)}$.
Step 6:  Take $c' = X'_{iR(LSB)}$ and calculate the syndrome $S' = (H \times (c')^T)^T$.
Step 7:  Concatenate syndrome S’ with data unit of D’ that is $D' = D' \| (S')$.
Step 8:  Repeat Steps 4 to 6 using $X'_{iG}$ and $X'_{iB}$.
Step 9:  Compute $D_i = D' \oplus k_{1.}$
Step 10: Repeat Steps 2 to 8 using next random sequence of $X_i$ block.
Step 11: Concatenate $D_i$’s, we get original secret message D.
Step 12: End.
```

3.1 Numerical Illustration

Example 1 Data Embedding

1. Let I is a color pixel block with (3 × 3) pixel. $D = \{d_1, d_2, \ldots, d_{36}\} = \{0, 1, 0, 1, 0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0\}$. $k_1 = \{0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0\}$ and ER = 36/(3 × 3) = 4 bpp and $D' = D \oplus k_1 = \{0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0\}$

2. Divided into three RGB pixel blocks shown below.

$$R=\begin{vmatrix}141 & 141 & 141\\141 & 142 & 142\\140 & 143 & 143\end{vmatrix}\quad \mathrm{G}=\begin{vmatrix}143 & 141 & 142\\141 & 144 & 145\\143 & 146 & 147\end{vmatrix}\quad \mathrm{B}=\begin{vmatrix}147 & 145 & 145\\145 & 149 & 150\\146 & 150 & 151\end{vmatrix}$$

3. Take red pixel block and transform into binary number matrix.

$$\mathrm{R}=\begin{vmatrix}10001101 & 10001101 & 10001101\\10001101 & 10001110 & 10001110\\10001100 & 10001111 & 10001111\end{vmatrix}$$

4. Divide it into 4 bit plane matrices starting from LSB.

$$R_{LSB}=\begin{vmatrix}1 & 1 & 1\\1 & 0 & 0\\0 & 1 & 1\end{vmatrix}\quad R_{LSB-1}=\begin{vmatrix}0 & 0 & 0\\0 & 1 & 1\\0 & 1 & 1\end{vmatrix}\quad R_{LSB-2}=\begin{vmatrix}1 & 1 & 1\\1 & 1 & 1\\1 & 1 & 1\end{vmatrix}$$
$$R_{LSB-3}=\begin{vmatrix}1 & 1 & 1\\1 & 1 & 1\\1 & 1 & 1\end{vmatrix}$$

5. Read the LSB matrix and form a 1D matrix. c = [1 1 1 1 0 0 0 1 1]
6. Calculate the syndrome

$$S_1=H\times(c)^T=\begin{vmatrix}0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0\\0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0\\0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1\end{vmatrix}$$
$$\times\begin{vmatrix}1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1\end{vmatrix}^T=\begin{vmatrix}1\\0\\1\end{vmatrix}$$

7. Transpose the syndrome and XOR with the secret data bit, i.e., [1 0 1] ⊕ [0 1 1] = [1 1 0] which matches with the fifth column of Hamming matrix.
8. Generate the code H' = [0 0 0 0 1 0 0 0 0] and XOR with the original code c.

   S_3 = [1 1 1 1 0 0 0 1 1] ⊕ [0 0 0 0 1 0 0 0 0] = [1 1 1 1 1 0 0 1 1].

9. Transform into a new LSB matrix.

$$R'_{LSB} = \begin{vmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{vmatrix}$$

10. Similarly compute the LSB-1, LSB-2, and LSB-3 matrices as follows:

$$R'_{LSB-1} = \begin{vmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{vmatrix} \quad R'_{LSB-2} = \begin{vmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \end{vmatrix} \quad R'_{LSB-3} = \begin{vmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{vmatrix}$$

11. Update all four modified binary matrices to their corresponding position in original RED pixel matrix.

$$R'_{3\times3} = \begin{vmatrix} 10001101 & 10001101 & 10001101 \\ 10000101 & 10001111 & 10001100 \\ 10001100 & 10001111 & 10001011 \end{vmatrix} = \begin{vmatrix} 141 & 141 & 141 \\ 133 & 143 & 140 \\ 140 & 143 & 139 \end{vmatrix}$$

12. Similarly get updated green and blue pixel matrices.

Example 2: Data Extraction

1. The marked frame pixel block is divided into three RGB color pixel blocks.

$$R = \begin{vmatrix} 141 & 141 & 141 \\ 133 & 143 & 140 \\ 140 & 143 & 139 \end{vmatrix} \quad G = \begin{vmatrix} 143 & 141 & 134 \\ 143 & 144 & 149 \\ 143 & 146 & 146 \end{vmatrix} \quad \mathrm{B} = \begin{vmatrix} 147 & 145 & 149 \\ 145 & 149 & 150 \\ 147 & 156 & 151 \end{vmatrix}$$

2. Take red image pixel block and transform into binary numbers.

$$\begin{vmatrix} 10001101 & 10001101 & 10001101 \\ 10000101 & 10001111 & 10001100 \\ 10001100 & 10001111 & 10001011 \end{vmatrix}$$

3. Divide it into 4 bit plane matrices starting from LSB.

$$R_{LSB} = \begin{vmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{vmatrix} \quad R_{LSB-1} = \begin{vmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{vmatrix} \quad R_{LSB-2} = \begin{vmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \end{vmatrix}$$

$$R_{LSB-3} = \begin{vmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{vmatrix}$$

4. Read LSB matrix and form a 1D matrix. c = [1 1 1 1 1 0 0 1 1]
5. Calculate the syndrome $S_1 = H \times (c)^T =$

$$\begin{vmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{vmatrix} \times \begin{vmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \end{vmatrix}^T = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$$

6. Transpose the syndrome to get secret data bits d = [0 1 1]
7. Repeat the above steps until we do not get the secret data bits. Concatenate all the data bits to get the data, that is, D' = {0, 1, 1, 1, 0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0}.
8. XOR the modified secret data with secret key k_1 to get the original secret data bits that is D = {0, 1, 0, 1, 0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0}.

4 Experimental Result and Comparison

The scheme is implemented using NetBeans IDE 8.0 on standard color images to measure the performance. The standard cover video sequences are Audio Video Interleave (AVI) format with the size of (192 × 352). These are collected from the video database of department of computer science at University of Mannheim [9]. The secret message is a binary image logo. Upon extraction, the secret data is retrieved without any loss or noise. The qualities of the stego frame images are measured using Mean Square Error (MSE) and Peak Signal-to-Noise Ratio (PSNR).

$$MSE = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} \left[I(i,j) - I'(i,j) \right]^2 \tag{1}$$

$$PSNR = 10 \log_{10} \frac{255^2}{MSE} (dB) \tag{2}$$

Following tables are representing the PSNR values of stego videos generated by our algorithm with varying payload. The average capacity of data embedding per frame is 16.5 KB (when ER = 2 bpp), 24.68 KB (when ER = 3 bpp), 32.9 KB (when ER = 4 bpp) which is almost eight times that of other existing algorithms [1]. The standard video sequences are 25 fps. So, we can embed maximum 823 KB data per second which is a high payload compare to any other algorithm [5] (Tables 1, 2, 3 and 4).

Table 1 Result of PSNR, SSIM from our proposed method when payload 2 bpp

	Akiyo	News	Cars	Train	Flamingo	Meer kat	Highway	Ice	Football
PSNR	50.17	50.17	50.18	50.18	50.17	50.18	50.17	50.18	50.17
SSIM	0.9936	0.9944	0.9928	0.9969	0.9932	0.9977	0.9966	0.9926	0.9976

Table 2 Result of PSNR, SSIM from our proposed method when payload 3 bpp

	Akiyo	News	Cars	Train	Flamingo	Meer kat	Highway	Ice	Football
PSNR	43.90	43.92	43.94	43.94	43.95	43.95	43.94	43.91	43.93
SSIM	0.9740	0.9776	0.9714	0.9876	0.9727	0.9908	0.9862	0.9705	0.9904

Table 3 Result of PSNR, SSIM from our proposed method when payload 4 bpp

	Akiyo	News	Cars	Train	Flamingo	Meer kat	Highway	Ice	Football
PSNR	37.81	37.82	37.87	37.86	37.93	37.88	37.87	38.04	37.85
SSIM	0.9065	0.9190	0.8985	0.9538	0.9020	0.9652	0.9501	0.8976	0.9636

Table 4 The performance comparison of different schemes under different video sequences

	Maximum capacity (kbits/s)			PSNR (dB)			SSIM		
Sequence	Xu et al.'s [10]	Wang et al.'s [3]	Proposed scheme	Xu et al.'s [10]	Wang et al.'s [3]	Proposed scheme	Xu et al.'s [10]	Wang et al.'s [3]	Proposed scheme
Stefan	17.8017	29.6121	6584	38.33	37.13	37.81	0.9825	0.9811	0.9665
Table	8.2683	12.7662	6590	37.91	37.39	37.82	0.9537	0.9526	0.9537
Mobile	1.8054	5.5506	6586	38.32	37.89	37.87	0.9871	0.9867	0.9567
Hall	0.5991	0.9696	6588	40.26	40.16	40.12	0.9743	0.9742	0.9788
News	0.4956	0.8181	6585	40.75	40.68	37.82	0.9848	0.9847	0.9190

5 Security Analysis

Security analysis is an important key factor of data hiding process. In this paper, we have used two levels of security to enhance our proposed scheme from security perspective. First, we take a 36 bits secret key and encrypt the secret data bits using symmetric key encryption. As it is only known to the sender and receiver, the third party will not be able to decrypt it without knowing the secret key. In the second level, we have taken a secret seed which is also known to the receiver and sender only. Using this seed, we generate a sequence of unique numbers with the help of PRNG function. We have taken the cover image pixel blocks according to the generated numbers. So without knowing this seed, no one will be able to predict the number sequence.

We also verified our algorithm against standard measurement like SSIM. The Structural Similarity (SSIM) index is a method for measuring the similarity between two images. The SSIM index can be viewed as a quality measure of one of the images being compared provided the other image is regarded as of perfect quality. From the tables, it is observed that the SSIM values of all test images are nearer to 1.

6 Conclusion

In this paper, we introduced a novel secure data hiding scheme using Hamming Code for video steganography. Bit plane slicing of the each RGB color cover frame pixel block is also introduced to increase data hiding capacity. So the data embedding rate is raised up to 4 bpp which is greater than other existing schemes [1, 4, 5]. In our algorithm, PSNR is also high compared to existing schemes [4, 5] which means that we generate better visual quality stego videos. From security perspective, we introduced a shared secret key to find suitable bit pattern through XOR operation during data embedding as well as data extraction. The cover video frame block has been chosen in random location through PRNG which enhances security. We have tested our stego image with SSIM and observed that the proposed scheme is preferable for data embedding where visual quality and security constraint needs to be maintained for high payload. In future, the scheme has been extended to enhance security, capacity, and quality in different domains for video-based steganography.

References

1. Mstafa, R.J., Elleithy, K.M.: A highly secure video steganography using Hamming code (7, 4). In: Systems, Applications and Technology Conference (LISAT), 2014 IEEE Long Island. IEEE (2014)
2. Cao, Z., Yin, Z., Hu, H., Gao, X., Wang, L.: High capacity data hiding scheme based on (7, 4) Hamming code. SpringerPlus **5**(1), 175 (2016)
3. Hu, S.D.: A novel video steganography based on non-uniform rectangular partition. In: 2011 IEEE 14th International Conference on Computational Science and Engineering (CSE). IEEE (2011)
4. Yadav, P., Mishra, N., Sharma, S.: A secure video steganography with encryption based on LSB technique. In: 2013 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC). IEEE (2013)
5. Dasgupta, K., Mandal, J.K., Dutta, P.: Hash based least significant bit technique for video steganography (HLSB). Int. J. Secur. Priv. Trust Manage. (IJSPTM) **1**(2), 1–11 (2012)
6. Banik, B.G., Bandyopadhyay, S.K.: Image Steganography using BitPlane complexity segmentation and hessenberg QR method. In: Proceedings of the First International Conference on Intelligent Computing and Communication (pp. 623–633). Springer Singapore
7. Fridrich, J., Du, R., Meng, L. Steganalysis of LSB encoding in color images. In: Proceedings of ICME 2000, Jul.–Aug. 2000, N.Y., USA
8. Westfield, A., Pfitzmann, A.: Attacks on steganographic systems. In: Proceedings of 3rd Info. Hiding Workshop, Dresden, Germany, Sept. 28–Oct. 1, pp. 61–75 (1999)
9. University of Mannheim, Department of Computer science. http://ls.wim.uni-mannheim.de/de/pi4/research/projects/retargeting/test-sequences/
10. Xu, W., Wang, R.D., Shi, Y.Q.: Data hiding in encrypted H.264/AVC video streams by codeword substitution. IEEE Trans. Inform. Forens. Secur. **9**(4), 596–606 (2014)