A Secure Storage and Transmission Method of Space Cloud Privacy Data

Yingzhao Shao(✉), Xiaobo Li, and Mingming Zhang

China Academy of Space Technology, Xi'an 710100, Shannxi, China
daisyshao1983@126.com

Abstract. With the rapid development of satellite Internet and space cloud, space cloud data security will become an important problem to be solved. In this paper, a method for secure storage and transmission of user privacy data in space cloud in the future is presented. By employing a large number of image data in the cloud, the Joint Photographic Experts Group (JPEG) image is decompressed into the quantized Discrete Cosine Transform (DCT) coefficients, and the mapping relationship between the DCT coefficients and the code stream is established. The maximum frequency of the non-zero coefficient is taken as the marked point, and the user privacy data is encrypted and embedded into each original code stream according to the Huffman code table. Finally, the final marked file with privacy data is obtained by merging the code streams of all code blocks. The marked file with embedded privacy data can be decompressed normally, and the decoded image can be displayed in a normal way. After the privacy data is extracted by legitimate users, the original carrier data can be recovered lossless. The experimental results show that this method ensures the secure storage and transmission of privacy data in the cloud without destroying the integrity and sharing of the original carrier data.

Keywords: Satellite internet · Cloud data security · User privacy data

1 Introduction

In recent years, innovative enterprises such as Space X and OneWeb in the United States have planned to build low-orbit satellite constellations, which has triggered an upsurge in the development of satellite Internet. Cloud computing is the third IT information revolution after the Internet. With the gradual construction of the satellite Internet system, space cloud will become an important space-based infrastructure. The United States has unveiled the "Space Belt" (SpaceBelt) Cloud Constellation program [2]. The space belt company plans to launch three geosynchronous orbit relay satellites and 10 data storage satellites at an altitude of 36,000 km from the earth. Among them, the data storage satellite will form a closed link (cloud constellation) in the low-Earth orbit, and communicate with each other and back up each other. A secure cloud server system with stable performance is installed on the data storage satellite, and through the communication between the

Q. Wu et al. (Eds.): WiSATS 2020, LNICST 358, pp. 502–513, 2021.
https://doi.org/10.1007/978-3-030-69072-4_41

synchronous orbit relay satellite and the special ground receiving station, the secure transmission of data in space and the secure storage of private data can be realized to meet the security requirements that can not be completely satisfied on the ground, such as preventing network attacks, data injection, data theft and so on. On March 22, 2019, Galactic Sky of the United States announced that its first software-defined satellite, GSky-1, has successfully completed the whole satellite integration in the Space Engineering Research Center (SERC) at the Information Sciences Institute (ISI) of the University of Southern California. The satellite uses Citrix virtualization technology, which will verify the functions of multi-customer satellite resource sharing, cloud node flexibility scalability, and cloud status monitoring and recovery in the cloud environment. On March 20, 2019, Lockheed Martin said it will test the space-based cloud computing infrastructure in the first star of the Pony Express project, which will eventually support the construction of a space cloud data center in the future. In addition, on November 20, 2018, Tianzhi-1, led by the Institute of Software of the Chinese Academy of Sciences, was launched from the Jiuquan Satellite launch Center and successfully entered the scheduled orbit of space. The satellite carries a "miniaturized cloud computing platform", which is mainly adopted to verify space-based cloud computing technology [3].

At present, almost all the data in cloud storage services are stored in clear text in the cloud, which is relatively simple to design, develop and deploy, and its users can easily share data. However, as this kind of cloud system can view all users' data directly, there is a great risk for the protection of users' private data. In order to protect the privacy and security of users' cloud data, it is a common method of encrypting the data and uploading it to the cloud server. However, this cloud data storage method poses a serious obstacle to data sharing, that is, the cloud server cannot directly send the ciphertext data of its owner to the sharer, because the sharer is incapable of decrypting the ciphertext that is not encrypted by his own key. In addition, due to the openness of space links, attackers are likely to steal or destroy cloud privacy or confidential data through illegal intrusion. How to strike a balance between the convenience of space cloud data sharing and private data security in the future is an urgent problem to be solved. Considering that a large number of cloud data are remote sensing image data, in order to solve the problem of secure storage and transmission of user privacy or confidential data, in this paper, a secure storage and transmission algorithm of privacy data for the cloud JPEG stream is proposed. Using this algorithm, the user privacy data is encrypted and hidden in the code stream corresponding to the highest frequency of non-zero coefficients for storage and transmission. For illegal users, even if the image data embedded with privacy data is obtained, it is difficult to find any anomalies; for shared users, the code stream embedded with privacy data can still be displayed with high quality after decompression, and does not affect the data sharing application; after the privacy data is extracted by legitimate users, the carrier image data can be restored in a lossless way for ensuring the integrity of the carrier data. With this method not only can the security of privacy data be ensured, but also the sharing of cloud space-based big data is satisfied. In this case, it has a broad application prospect in cloud private data storage and transmission.

2 Related Works

At present, there are four kinds of data embedding techniques based on JPEG format: re-quantizing DCT coefficients [4–6], modifying the quantization table [7, 8], modifying the Huffman coding table [9–12], and embedding data in encrypted JPEG code streams [13, 14]. These methods well adapt to the rules of JPEG coding and improve the application range of JPEG image data embedding, but most of the embedded images can not be shown normally, or the information of the carrier file is lost.

Wang et al. [7] first proposed the algorithm to achieve high-quality display and large embedding capacity, and the file expansion is also limited. He not only modified the quantized DCT coefficients, but also changed the quantization table. Although the expansion has been restrained, the expansion problem is still very serious, and it is not suitable for the case that QF is 100. Huang et al. [8] proposed a histogram shift algorithm for DCT coefficients of JPEG images. Coefficients with values of −1 and 1 in each DCT block are used to embed secret information, and a block selection strategy is added to further improve the capacity with the number of zeros in DCT blocks. However, there is a certain deviation between the theoretical value and the actual value of this method, and the expansion is also very obvious. Hou et al. [9] optimized Huang et al.'s [8] algorithm, that he selects K frequency points with the smallest deviation in the DCT block, and shifts the coefficients of the frequency points whose values are -1 and 1. In the same embedding capacity, it has better image quality and less expansion. Liu et al. [10] proposed a simple and efficient algorithm in which all non-zero coefficients in each DCT block are changed to embed data, such that capacity is greatly improved and the file expansion is relatively small. Zhang et al. [11] proposed an adaptive hiding algorithm for JPEG code stream, which truncates the code stream corresponding to high-frequency coefficients and hides information with spare space. The marked images can be displayed normally, but the original data is lost to some extent. Zhang et al. [12] proposed a lossless hiding algorithm for JPEG code stream, which recompresses the high-frequency coefficients to hide secret information. The algorithm can restore the image lossless, but the modified file can not be decompressed normally in the process of transmission.

3 The Proposed Method

In order to ensure the secure transmission of JPEG remote sensing images in space cloud, an information embedding algorithm is proposed in this paper. Secret information is preprocessed and put behind the code stream corresponding to the non-zero coefficients with the highest frequency, so that the marked images can be displayed with high quality. This method ensures both the security of secret information and the integrity of image data.

In the Huffman code table of JPEG, Run/Size is the joint coding of Alternating Current (AC) coefficients. For each non-zero coefficient, Run represents the number of coefficients before the non-zero coefficient is 0; Size indicates the binary coding length of the non-zero coefficient; Value refers to the value of the non-zero coefficient; code length is the length of Run/Size joint coding; Codes are the corresponding codeword, and the binary coding length of Variable Length (VL) is the sum of code length and VL. In

other words, DCT coefficient coding is the joint coding concerning the number of non-zero coefficients and non-zero coefficients. For example, in the coefficients sequence {0,0,0,4}, Run is 3, Size is 3, the corresponding Run/Size code is "111111110101", added with the binary code "100" of 4, and the joint coding of the sequence {0,0,0,4} is "111111110101,100" (Table 1).

Table 1. Huffman coefficient encoding

Run/Size	Value	Code length	Codes	Sum length	VL
0/0(EOB)	0	4	1010	4	0
0/1	−1, 1	2	00	3	1
0/2	−3, −2, 2, 3	2	01	4	2
0/3	−7, −6, −5, −4, 4, 5, 6, 7	3	100	6	3
0/4	−15, −14,…,−8, 8,…,14, 15	4	1011	8	4
0/5	−31, −30,…,−16, 16,…,30, 31	5	11010	10	5
1/1	−1, 1	4	1100	5	1
1/2	−3, −2, 2, 3	5	11011	7	2
1/3	−7, −6, −5, −4, 4, 5, 6, 7	7	1111001	10	3
1/4	−15, −14,…,−8, 8,…,14, 15	9	111110110	13	4
2/1	−1, 1	5	11100	6	1
2/2	−3, −2, 2, 3	8	11111001	10	2
2/3	−7, −6, −5, −4, 4, 5, 6, 7	10	1111110111	13	3
3/1	−1, 1	6	111010	7	1
3/2	−3, −2, 2, 3	9	111110111	11	2
3/3	−7, −6, −5, −4, 4, 5, 6, 7	12	111111110101	15	3

In each DCT block, as the value of the high frequency coefficient is small and sparse, a small modification to the high frequency coefficient will not cause great attenuation of the image quality. In order to ensure the blind extraction of secret information at the receiver and the unchanging original coefficient, it is necessary to find a suitable high-frequency termination point T and embed secret data in the high-frequency termination point. When the value of the non-zero coefficient exists in $\{-3, 2, 2, 3\}$, the corresponding coding length is 2 bits, and its effect on image quality at high frequency is negligible. Corresponding to the selected termination point T, it should be ensured that there is no

non-zero coefficient above it. Below this point, there are several zero coefficients, and the number of zero coefficients is M. In this paper, the 2-bit mapping for secret data can be established, as shown in Table 2. For example, when the secret data is "00", the non-zero value at point T is −3. Then, the final bit data is obtained according to Table 3. In this case, the Size is all 2, or the coding length of non-zero coefficients is 2 bits, and Run/Size can be regarded as Run/2. Table 3 shows a simplification of the Huffman code table, and its coding process is fully in line with the JPEG coding standard. When T is less than 64, all the high frequency coefficients above T are 0, and it is necessary to add "1010" for indicating the end of coding. For example, when T is 53, secret data is "01", and the number of zero coefficients Size below the high frequency point T is 3, then the non-zero value in the point T is changed from 0 to −2, and the joint coding is "111110111,01". With "1010" as the ending identifier End Of Bits (EOB), the changed code stream is "111110111,01,1010". The coefficients below T are not changed as well as the corresponding code stream, indicating that there is no harm to the original low-frequency coefficients, and the carrier data can be recovered lossless after the secret data at the receiver is extracted completely. The cost is a certain degree of file expansion, which is negligible compared with the protection of secret data.

Table 2. Relationship between secret data and value

Secret data	00	01	10	11
Value	−3	−2	2	3

Table 3. Joint coding table of secret data

Run	Value			
	−3	−2	2	3
0	01,00	01,01	01,10	01,11
1	11011,00	11011,01	11011,10	11011,11
2	11111001,00	11111001,01	11111001,10	11111001,11
3	111110111,00	111110111,01	111110111,10	111110111,11
4	1111111000,00	1111111000,01	1111111000,10	1111111000,11
5	11111110111,00	11111110111,01	11111110111,10	11111110111,11

In this way, the changed DCT coefficient at the highest frequency T point is limited to $\{-3, -2, 2, 3\}$, which not only guarantees a certain hiding capacity, but also maintains the high quality display of the marked image or the invisibility of secret data. The PSNR of the marked image is kept above 35 dB, and the distortion between the marked image and the original JPEG image is extremely small. Therefore, it is difficult for human eyes to distinguish. In each block, the high-frequency termination point T is adaptive, and its value is the corresponding highest frequency of the original non-zero coefficient added

with 1. When the original non-zero highest point is 64, it means that there is no additional frequency for embedding secret data. In order to ensure the generalization ability of this algorithm, the maximum frequency corresponding to the original non-zero coefficient is 62. In this case, the high-frequency termination point T is 63. When there is a non-zero value at frequency 63 or 64 in the original DCT block, the code block cannot hide the data. In extreme cases, as shown in Table 4, even if there are non-zero values in the original frequencies 62 and 63, and the values are in {−3, −2, 2, 3}, a value of 7 can be selected as a marker on frequency 64. According to a large number of experimental statistics, in the highest frequency 64, only a few non-zero values are −2 or 2. When the highest frequency of the non-zero coefficient is 62, and its value is in {−3, −2, 2, 3}, and the value of the non-zero value of the secret data in frequency 63 is at {−3, −2, 2, 3}, the coefficient in the frequency 64 is set to zero, so that whether the code block is embedded in the secret message can be judged by the coefficient value of the highest frequency 64. In this way, it is possible to determine whether the code block is embedded in the secret data by the coefficient value of the highest frequency 64. When the coefficient in the 64 frequency point is zero, the secret data is embedded, while when its value is not zero, the secret data is not embedded.

Table 4. Extreme case handling table

Embedding situation	Frequency		
	62	63	64
Original coefficient	−3, −2, 2, 3	−3, −2, 2, 3	7
Embedding coefficient	−3, −2, 2, 3	−3, −2, 2, 3	0

The data embedding process of an 8 × 8 DCT block is shown in Fig. 1. The original DCT block is displayed in Fig. 1(a); the highest frequency corresponding to its non-zero value is 31, and the frequency diagram is shown in Fig. 1(b); 32 is taken as the high-frequency termination point T, and the data is embedded at this point, assuming that the secret data is "10", while the corresponding value is 2, and the marked DCT block is shown in Fig. 1(c). It can be seen that the information on the DCT block is changed only slightly, and the distortion of the image corresponding to the spatial domain is very small. In terms of the code stream, the code stream below frequency 31 has not changed. Above this point, the code stream changes from "1010" to "01101010", where "0110" is the encoding with a Value of 2, and "1010" indicates the end of the encoding.

At the sender in the cloud, the data embedding can be divided into four steps, as shown in Fig. 2.

1. Decompress the JPEG file to the DCT blocks, and establish the corresponding relationship between the DCT coefficients and the code stream.
2. Make statistics of each DCT block which can be embedded in data to get the final embedding capacity.

12	17	10	-5	0	-2	0	0
15	8	0	0	0	0	0	0
-9	0	0	0	0	-1	0	0
-3	4	0	0	0	0	0	0
0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

(a)

1	2	6	7	15	16	28	29
3	5	8	14	17	27	30	43
4	9	13	18	26	31	42	44
10	12	19	25	32	41	45	54
11	20	24	33	40	46	53	55
21	23	34	39	47	52	56	61
22	35	38	48	51	57	60	62
36	37	49	50	58	59	63	64

(b)

12	17	10	-5	0	-2	0	0
15	8	0	0	0	0	0	0
-9	0	0	0	0	-1	0	0
-3	4	0	0	0	0	0	0
0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

(c)

Fig. 1. An 8 × 8 DCT block data embedding schematic diagram

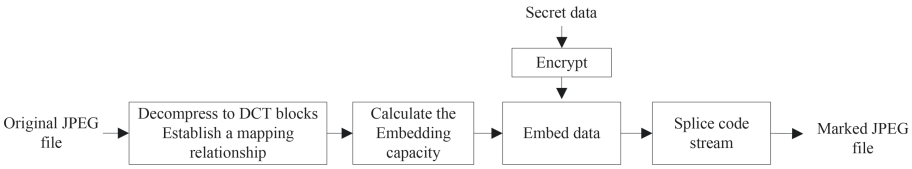

Fig. 2. Data embedding flowchart

3. The secret data is encrypted and embedded in the DCT blocks with a set of 2 bits according to the proposed coding algorithm.
4. All the changed code streams are spliced together to form a marked JPEG file.

At the receiver in the cloud, the data extraction and recovery can be divided into four steps, as shown in Fig. 3.

1. Decompress the JPEG file to the DCT blocks, and also establish the corresponding relationship between the coefficients and the code stream.
2. Judge the utilization of each block according to Table 4.
3. For the utilized code blocks, the coefficient corresponding to the highest frequency is decoded into a bit stream, and the coefficient is set to 0; for the unused code block, the original coefficient is repaired according to the coefficient at the 64 frequency point.

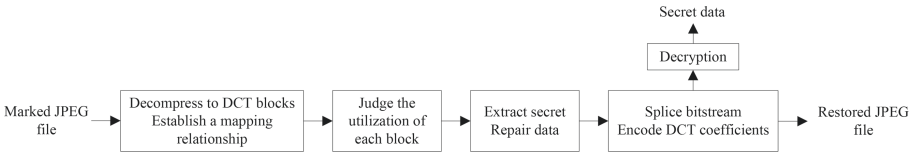

Fig. 3. Data extraction and recovery flowchart

4. Continue to encode the repaired coefficients to get the final recovered compressed file. The extracted bitstream is spliced and decrypted to get the final extracted data.

4 Experimental Results

In this paper, 38 images in the dataset of Southern University are adopted to verify the algorithm, and the test images are all lossless TIFF grayscale images with the 512×512 format. When the QF is greater than 20, the visual quality of the image is higher. Thus, the QF is set as 30, 50, 70 and 90 to compress the image. The secret data is a random binary bit stream. The key data is shared in the sender and the receiver, and the key data is a pseudo-random seed. In the sender, a pseudorandom 0,1 sequence is generated, and it is XOR with the original secret data. In addition to that, the encrypted secret data is obtained. In the receiver, the same sequence is XOR with the extracted secret data, and the final secret data is gained.

The experimental platform is MATLAB 2013a; the operating system is 64-bit windows 7; the CPU is i5-5200u; the main frequency is 2.2 GHz, and the memory is 8 GB. Peak signal-to-noise ratio (PSNR) and embedding capacity (EC) are employed to evaluate the performance. PSNR refers to the value of the marked JPEG image relative to the original JPEG image. It is worth noting that the higher the PSNR, the better the image quality and the invisibility of secret data, while the higher the EC, the stronger the ability of the image to carry information.

The EC and PSNR of the first 10 remote sensing images are shown respectively in Table 5 when the QF is 30, 50, 70 and 90. When the QF is small, the quantization step is large, and a large amount of data is removed, so that the coefficient on the high frequency is extremely small. For the 512×512 image, there are a total of 4096 blocks, each of which can be embedded with 2 bits. Hence, in theory, up to 8192 bits of information can be embedded. With the increase of the QF, the quantization step decreases and a large amount of data is retained. In this case, the coefficients in high frequencies become larger and denser, and some blocks that fail to meet the coding of this paper appear, but the number of these blocks is very small and can be ignored. When the QF is small, the retained coefficient is less. Although the change is the same, the proportion of the existing DCT coefficient is larger, and the PSNR value is lower. With the increase of the QF, more coefficients are kept, and these slight changes are smaller than the existing DCT coefficient, and the PSNR value is larger. However, on the whole, the PSNR values are all around 40 dB, and the human eyes are unable to distinguish the distortion.

When the QF is 30, 50, 50, 70 and 90, the comparison between the 2nd and 10th images is shown in Fig. 4 and Fig. 5. Figure (a) is the original JPEG image with QF 30; (b) shows the original JPEG image with QF 50; (c) displays the original JPEG image with QF 70; (d) is the original JPEG image with QF 90; (e) is the marked JPEG image with QF 30; (f) is the JPEG image with QF 50; (g) shows the JPEG image with QF 70, while (h) displays the JPEG image with QF 90. It can be seen in Table 6 that the texture of the second image is complex; the PSNR value of the same QF is lower; there are a large number of non-zero high-frequency coefficients; the texture of the 10th image is relatively simple; the PSNR value of the same QF is higher, and there are a small number of non-zero high-frequency coefficients. However, whether the simple image

Table 5. EC and PSNR of the first 10 remote sensing images

Number	QF							
	30		50		70		90	
	EC (:bits)	PSNR (:dB)	EC (:bits)	PSNR (:dB)	EC (:bits)	PSNR (:dB)	EC (:bits)	PSNR (:dB)
1	8192	37.02	8192	38.10	8192	38.12	8188	41.69
2	8192	34.72	8192	37.73	8190	36.93	8188	39.20
3	8192	41.08	8192	43.44	8192	44.35	8186	48.00
4	8192	37.14	8192	38.47	8192	38.83	8188	42.26
5	8192	38.32	8190	41.37	8190	41.74	8184	45.89
6	8192	37.42	8192	39.48	8192	40.11	8190	43.68
7	8192	39.56	8192	40.87	8192	41.57	8190	46.28
8	8192	40.44	8192	44.50	8190	46.03	8186	57.04
9	8192	41.59	8192	43.83	8192	46.08	8190	52.50
10	8192	38.91	8192	40.84	8192	41.90	8188	45.68

or the complex image, the distortion amplitude of the image is still within the range acceptable to the human eyes, or in other words, the human eyes fail to distinguish the distortion.

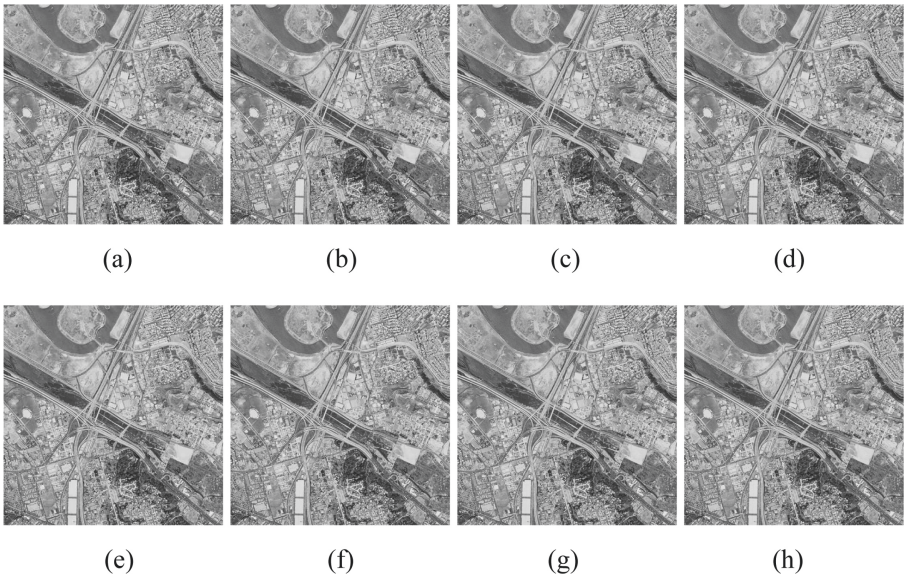

Fig. 4. Embedding effect of the 2nd remote sensing image

In order to further elaborate the effectiveness of this algorithm, in this paper, EC and PSNR are compared, with Wang et al.'s [7], Huang et al.'s [8], Hou et al.'s [9] and Liu et al.'s [10]. These algorithms can guarantee that the image can be restored lossless in the receiver, and the hidden marked image can be displayed normally, while the cost is more consistent, which will cause file expansion to a certain extent. The average EC comparison of 38 images under different QFs is shown in Fig. 6, and the average PSNR comparison is displayed in Fig. 7. It can be seen that the EC of this paper is relatively low, while the PSNR is relatively high, because the modified DCT coefficient of this algorithm only corresponds to a separate high frequency point, and the change appears in a small range. In algorithm [7], Wang et al. embed data by modifying quantization step size. In algorithm [8], Huang et al. embed data through DCT block histogram shift, while in algorithm [9], Hou et al. embed data by selecting frequency and blocks. The frequency point which is most suitable for hiding is selected to embed the data. In algorithm [10], Liu et al. multiply the whole coefficients to obtain redundant space to embed data. These algorithms modify the coefficients in a large range of DCT blocks, and the corresponding EC is also relatively large, but the image quality is not high. In this way, even though the marked image can be displayed, there will be modified traces, which can be easily found, and the security of the image can not be satisfied. In this paper, we make a compromise between EC and PSNR, lay emphasis on PSNR, make the secret data in the marked image with high invisibility, and carry a certain amount of secret data.

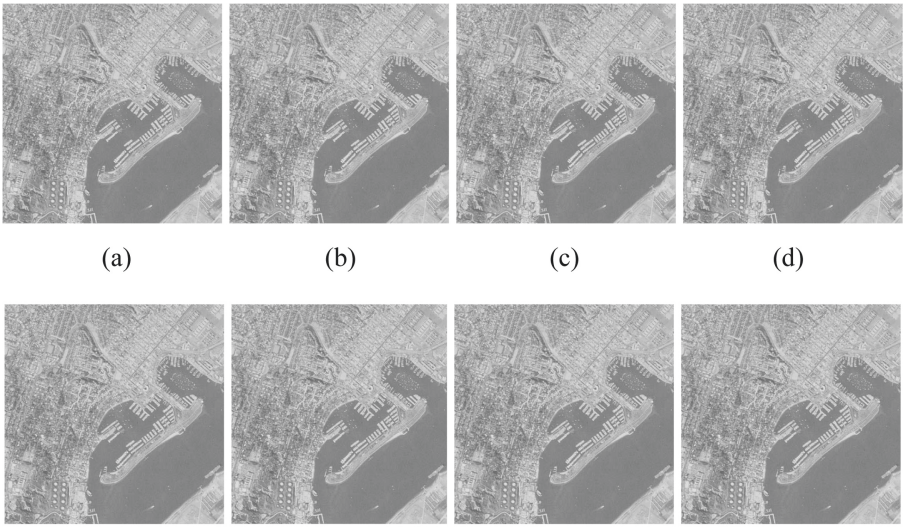

Fig. 5. Embedding effect of the 10th remote sensing image

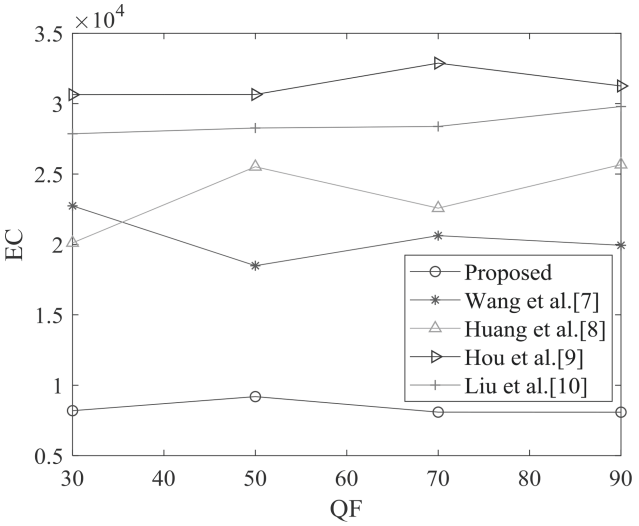

Fig. 6. Average EC comparison (unit: bits)

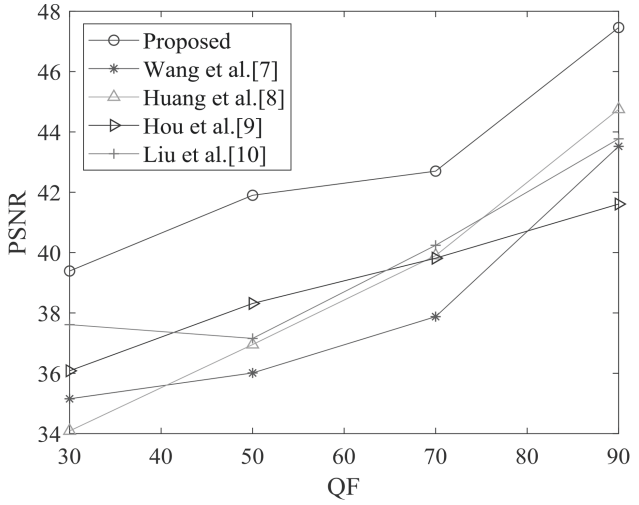

Fig. 7. Average PSNR comparison (unit: dB)

5 Conclusions

Aiming at the contradiction between the convenience of data sharing in space cloud and the security of privacy data in the future, this paper proposes a method based on the secure storage and transmission of privacy data belonging to space cloud users in the future. By using a large amount of image data in the cloud, the user privacy data is embedded in the image stream for secure storage and transmission. On the premise of ensuring the integrity and sharing of the original carrier data, it meets the requirements

of the secure storage and transmission of private data in the cloud. Furthermore, it can be taken as a reference for solving data security problems in open and shared environments such as ground clouds and future space-based clouds.

References

1. Gao, Y.Y., Wang, N.W., Lu, Z.: The development research and construction suggestion of satellite internet constellations. J. China Acad. Electron. Inf. Technol. **14**(8), 875–881 (2019)
2. Cloud Constellation Corporation Selects LeoStella to Manufacture the SpaceBelt Constellation. https://www.parabolicarc.com/tag/spacebelt/. Accessed 12 Nov 2019.
3. China's first software-defined satellite, Tianzhi-1, was successfully launched. https://www.cas.cn/cm/201811/t20181121_4671546.shtml. Accessed 20 Nov 2018
4. Huang, F.J.: Reversible data hiding in JPEG images. IEEE Trans. Circuits Syst. Video Technol. **26**(9), 1610–1621 (2016)
5. Hou, D.D.: Reversible data hiding in JPEG image based on DCT frequency and block selection. Signal Process. **148**(10), 41–47 (2018)
6. Liu, Y.J., Chang, C.C.: Reversible data hiding for JPEG images employing all quantized non-zero AC coefficients. Displays **51**(2), 51–56 (2018)
7. Wang, K., Lu, Z.M., Hu, Y.J.: A high capacity lossless data hiding scheme for JPEG images. J. Syst. Softw. **86**(7), 1965–1975 (2013)
8. Di, F.Q.: Reversible data hiding in JPEG images based on zero coefficients and distortion cost function. Multimedia Tools Appl. **78**(24), 34541–34561 (2019)
9. Hu, Y.J., Wang, K., Lu, Z.M.: An improved VLC-based lossless data hiding scheme for JPEG images. J. Syst. Softw. **86**(8), 2166–2173 (2013)
10. Qiu, Y.Q.: Lossless data hiding in JPEG bitstream using alternative embedding. J. Vis. Commun. Image Represent. **52**(2), 86–91 (2018)
11. Qian, Z.X., Zhang, X.P., Wang, S.Z.: Reversible data hiding in encrypted JPEG bitstream. IEEE Trans. Multimedia **16**(5), 1486–1491 (2014)
12. Nasrullah, N.: Reversible data hiding in compressed and encrypted images by using Kd-tree. Multimedia Tools Appl. **78**(13), 17535–17554 (2019)
13. Zhang, M.M., Zhou, Q., Hu, Y.L.: A reversible data hiding scheme in JPEG bitstreams using DCT coefficients truncation. KSII Trans. Internet Inf. Syst. (TIIS) **14**(1), 404–421 (2020)
14. Zhang, M.M., Zhou, Q., Hu, Y.L.: Lossless data hiding in JPEG images with segment coding. Journal of Electronic Imaging (JEI) **28**(5), 053015(1–14) (2019).