**Responses to the reviewers’ comments**

Dear Editor and Reviewers:

Thank you very much for your time in reviewing our manuscript “Secure image sharing over unsecured channels" and giving us the opportunity to revise the manuscript. We have now revised our manuscript in response to all the reviewers' comments. The revised manuscript along with a response document is resubmitted for your second review.

Thanks and best regards,

Authors

**Reviewer #1:**

Comment 1: Table 1 looks naive with lengthy descriptions as cell values. Either make the entries shorter or remove this table. Only literary descriptions would be much better than that kind of table. Also, it's kind of weird to make a comparison even before describing own proposal! It would be much better to present the related works in a logical way and then to put own motivation for the new or enhanced or changed approach.

Response 1: We greatly appreciate the comment. To address this comment, we removed the table and added the information below in the text at the end of the ‘past work’ section, 2.2:

“Yuan and Xu [24, 25] proposed to encrypt the image using a symmetric encryption scheme such as the chaos-based technique or one-time pad and distribute the encrypted image with the key to different participants. However, the shares should be the same size as the original image. Therefore, this approach requires a great amount of time to achieve effective encryption. Shi [26] proposed to encrypt the secret image by generating public and private images, then distributing the public image to the participants with the private image which used SSS. However, the problem of increasing the share size between the participants will arise, which will decrease the speed and require a great amount of time to encrypt and decrypt the images. Lang and Malik’s [4,27] method was to encrypt the image using any encryption scheme then distribute the encrypted image to all the participants without exchanging the key by using a three-pass protocol or with a keyless approach. This minimizes the information shared between the dealer and the participants. However, the information can be leaked as decryption is possible for the chain of users. Feldman [28] proposed to share the image using SSS then encrypt each share before distributing them among the participants using a public key. Still, all the data must be encrypted before sending the shares over unsecure channels, which will be time-consuming in encrypting and decrypting.

In contrary to the above works, the proposed method encrypts only a part of the image depending on the number of shares and then shares the images with the encrypted keys. This minimizes the size of the shares and therefore reduces the execution time for sharing and retrieving the image, while achieving security on unsecured channels.”

Comment 2: A clear vision for future gains from the work or possible applications in real-life cases are not understood from the work. Please include some more convincing possible future directions of research based on the work. It should be discussed why and how the numerous other alternatives are not enough to satisfy the application requirements. A bit clarity is needed.

Response 2: Thank you for the comment. To address this comment, we have revised the second paragraph in the introduction section, as follows:

“Many researchers [1-15] discussed methods for sharing an image or data among different users. These methods assume that an attacker cannot gather *t* or more shares. This assumption may not hold true especially in unsecured networks such as a wireless camera network. We found that there is little research focusing on sending shares over such unsecured networks in a trusted and safe way, such that even if an attacker finds *t* or more shares, s/he should not be able to obtain any information about the secret data. The main focus of this paper is to deal with security issues in such scenarios.”

Comment 3: The attacks that are thwarted by the scheme should be mentioned by name (each one separately to analyze those) and it should be discussed why and how those are deterred. The security analysis section is short and could be expanded a bit to put more convincing arguments

Response 3: Thank you for the suggestion. Accordingly, we have added a new subsection (5.2) in section 5 (see below). What used to be subsection 5.2 has become subsection 5.3.

“5.3 Sharing over unsecure channels

The proposed method achieves the following goals which were not met by the previous methods:

* Sending the image shares in a secure way.
* Enhancing and improving the speed of the system by encrypting part of the image.
* Keeping the main concept of the image sharing without modifying it.

The security of shared images is analyzed using the following two propositions. The assumed parameters are:

* S: Secret data
* R: Random values
* O: Original Data
* S-space: The attacker can sniff no more than n-1 shares over the channels.
* U-space: The attacker can sniff more than or equal to n shares over the channels.

*Proposition 1: The SSS which uses one secret value and random numbers is secured over S-space and unsecured (can be attacked quickly) over U-space.*

Proof: The SSS is secured over the S-space since the attacker can't retrieve information if less than n-1 is used, whereas in U-Space the attacker can retrieve the information because all the shares are used in the retrieving process.

*Proposition 2: The proposed method is secured over S-space and U-space.*

Proof: The proposed method contains three phases. In the first and second phases, the image is divided into parts, which are used to create a linear independence between parts. Then, the first part of the image will be encrypted to hide the solver information. In the last phase, the data is shared. In addition, after the final sharing in phase 3, the pixel values are changed completely because the replacement, sharing and the linear independence relationship (in phase.2) are used, which randomize the pixels in an effective way. Mathematically, this phenomenon can be explained as follows.

1. Replacement (O) 🡪 RO
2. Sharing(RO) 🡪 SO
3. Linear independence (SO) 🡪 LO
4. Shuffling(First Part in LO) 🡪 SLO

Therefore,

Finally, phase 3 shares the data with the participant using the following equation.

The proposed method used three steps to hide the information and to increase the randomness in the output data, so the original data (secret data) is changed to random space (R) rather than the original space (S).”

**Reviewer #2:**

Comment.4: The paper presents a proposal for sharing secret images based on Shamir's secret sharing (SSS) scheme, where the input image is split into equal-size parts, the first of which is encrypted using chaotic maps and linearly combined with the others, and finally shares are created from the resulting parts. The images protected are presumably graylevel only, as the text does not mention colour at all, and all examples are graylevel images.

Response 4: Thank you for the comment. We have added the following statement in the first paragraph of section.4:

“For the color images the process will be the same but instead of using only one matrix (for the gray channel), three metrics will be used (one for each color channel). Therefore, in all of the below processes the gray images will be used.”

Comment 5: While the paper takes some effort to show the randomness and visual unrecognisability of the generated shares, it does not show how the shares can reconstruct the original image, and in this reviewer's opinion such reconstruction is impossible thereby making the proposal unusable.

Response 5: Thank you again for the comment. We agree that the details of the reconstruction method should be provided in the paper. Accordingly, we have added a new subsection (4.4, see below) in the experimental section. Furthermore, the reviewer can read the flowchart in figure.3 to follow the logical order of the reconstruction phase.

4.4. Reconstruction Phase

To reconstruct the image, the minimum number of participants should collaborate together, using Lagrange interpolation to retrieve the parts of the image. The first part is then decrypted using the reverse order of the encryption algorithm after generating the chaotic values. The decrypted part solves the linear independence relationship with the other parts using the equation below.

Partno(r, c) =( Partno(r, c)- Part1(r, c) ) mod 251 2, 3…t1

After that, the resulting parts are used to create the original image by using Lagrange interpolation and row and column replacement using the equation below. Finally, the padded pixel is removed.

I(i, j) = ( (I(i, j) – ((i + j) mod 251) ) mod 251.

For brevity, we used the cameraman image with t=3 and t=6, respectively, where the number of parts is equal to the number of shares as shown in Figure 9.

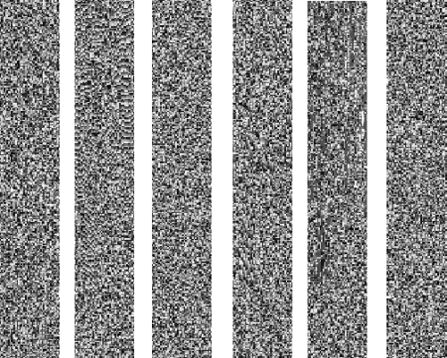
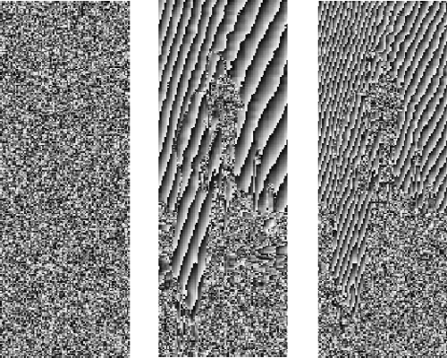




Figure.9: Reconstruction phase for the cameraman image using two different shares and part numbers

Comment 6:  The use of SSS in two stages of the process is confusing and not justified.  In the manuscript there is a mix-up between image parts and image shares and the relationship between them is unclear.  Eq. (8) adds to this confusion, as it seems to imply that the number of final image shares t\_2 must equal the number of image parts t\_1, and that shares must be the same size as parts.  Indeed t\_2 < t\_1 would not make sense as it would be impossible to reconstruct t\_1 parts from a smaller number of shares using SSS.  But in section 4 there are examples with t\_1 != t\_2, and even in subsection 4.4.3 some tests are made with t\_1 > t\_2 = 3.

Response 6: We appreciate the comment. To clarify this point, we have added the following text at the beginning of subsection 4.5 (Number of Shares)

“The part is the image share after applying SSS in phase.1, where the share is the image share after applying SSS in phase 3 as shown in Figure 3. Also, t1 and t2 are parameters used to denote the number of parts and shares, respectively. The parts are used to create a linear independence relationship with other parts. In order to increase the randomness of the proposed method, the parts will then be combined to create a new image. Phase 3 uses the new image as input to create shares using SSS. Therefore, the relationship between two parameters is that increasing the number of parts increases the randomization and enhances the shares in phase 3.”

Comment 7: In the first stage of the proposed method there is a replacement of pixel values to a range between 0 and 250.  This is unjustified and arbitrary.  Furthermore, it makes the algorithm irreversible.  If the input pixel range is 0-255, values 251-255 are indistinguishable from 0-4.  Therefore, the 5th step in Algorithm 2 ("Replacing Image pixels using reverse Eq. 3") does not make sense as Eq. (3) cannot be reverted.  (Incidentally, the first "mod 251" in that equation is superfluous.)

Response 7: We appreciate the comment, however we would like to clarify that using “mod 251” is a standard practice in SIS methods. Furthermore, the number of pixels that are greater than 250 in standard images is small, therefore truncating the pixels that are greater than 250 creates no noticeable effect.

We added the following text in subsection 3.1.1 (Phase 1 - creating parts)

“Using “mod 251” is a standard practice in SIS methods. Furthermore, the number of pixels that are greater than 250 in standard images is small, so truncating the pixels that are greater than 250 creates no noticeable effect.”

Comment8: The performance of the share generation process is measured in different aspects, but nothing is said about image reconstruction, which does not seem to have been tested.  (As explained above, image reconstruction looks impossible to this reviewer's understanding.)

Response 8:Thank you for the comment. We have addressed this in the response to comment 5.

Comment 9: Fig. 2 and Fig. 3 are interchanged:  Fig. 2 should be Fig. 3 and Fig. 3 should be Fig. 2 (which would make the reference to "three phases" in the last sentence of subsection 3.3 inapplicable).  Also, a reference to Fig. 4 in subsection 4.1 should refer to Fig. 5.

Response 9: Thank you for pointing out these errors. We have corrected them.

Comment10: Although not relevant to the algorithm description it seems that, in the notation used in the manuscript, coordinate pair (i, j) refers to row i (y-coordinate), column j (x-coordinate), i.e. coordinates are given as (y, x) or (row, column), rather than (column, row).

Response 10:We believe that the (y,x) format (e.g. Pair(i,j)) can be better understood for programming purposes.

Comment 11: There is a notational mix-up in some parts of the manuscript, e.g. at the start of subsection 3 values w and h are immediately renamed as N and M (or M and N?), and then as m and n in Eq. (5).  Also, what is called n in the 1st line of Eq. (5) is then called t\_1 in the 2nd line.  In the same Eq. (5) the relationship between (r,c) and (i,j) is not specified:  it is assumed that i = r, j = (no - 1) \* m / t\_1 + c.

Response 11: Thank you for pointing it out. We have corrected it.

Comment 12: the keys key\_1 and key\_2 of Eq. (7) are used in Eq. (1) is not specified.

Response 12: We have added the following text at the end of subsection 3.2:

“Moreover, after generating key1 and key2 using Eq.7, the two keys will be used in Eq.1 separately to generate x-values and y-values, i.e. to generate x-values x0=key1 and to generate Y-values x0=key2.”

Comment 13: The specification of algorithms 1 and 2 referenced in subsections 3.2 and 3.3 are extremely sketchy.

Response 13: Thank you for the comment. We have revised the description of the algorithms by adding step numbers.

Comment 14: Image encryption, applied only to the first part, seems to make use of small keys (e.g. in the experimental part, the values 1234 and 10 are used), which renders brute-force searches very easy.  However, subsections 4.5.1 and 5.1 mention a key space of size 10^15.  Where does that key space come from?

Response 14: To clarify this point, we have added the following text in the first paragraph of section 4:

“Note that in although in experiments we have used 1234 and 10 as keys, in the real scenarios the key and the factor will be greater than these values. The key used in our method consists of a ﬂoating-point number. If we use the ﬁrst 15 digits after decimal, the possible number of keys will be 10^15.”

Comment 15:  It is assumed that the notation "share 1", "share 2", etc. used in the experimental part refers to the image obtained by applying Eq. (5) with x = 1, x = 2, etc.

Response 15: Yes, that is right. We have further clarified it by adding the following text at the end of section.4 before section 4.1

“To generate the shares using Eq. 2, any x values less than 250 can be used. For example, for share1 we can use x=40, and for share2, x=50. For simplicity, assume that share 1will use x=1 and share2 will use x=2, etc."

Comment 16: Fig. 8 deserves some comment.  If the original image is uniformly white, why is part 1 in phase 2 not uniform?  
  
Response 16: This is due to the chaotic encryption which is applied only to the first part, as described in section 3.1.1 paragraph 6.

The following text has been added to Section 4.3:

“Note that, in Figure 8, even though the original image is uniformly white, part 1 in phase 2 (Figure 8a) is not uniform. This is due to the chaotic encryption which is applied only to the first part.”

**Reviewer #3:**

Comment 17: It would be better if authors perform analysis about computational complexities with proposed algorithm. It would be necessary if authors provide statistical performances about security analysis in Sec. 5.

Response 17: Thank you for the comment. In the revised version, we have added a subsection (5.4) as below:

5.4 Computational complexity

The computational complexity of the proposed method is O(n2), where n2 is the size of the image. Various steps to analyze the computational complexity are provided in the table shown below.

|  |  |
| --- | --- |
| Algorithm part | Description / Number of nested loops |
| Replacement | The pixel’s image is replaced and changed to a new value. Two nested loops depend on the image size. |
| Sharing | The image uses SSS to generate the shares. Two nested loops depend on the image size. |
| Linear independence | The first share is used to create a linear independence relationship with the other parts. Two nested loops depend on the image size. |
| Shuffling | The first part is shuffled using the chaotic keys. Two nested loops depend on the M/t1 \* N. |
| Sharing | The image is combined and then SSS is used. Two nested loops depend on the image size. |

Comment 18:

Authors are suggested to refer to the following papers in the reference to broaden the scope of this submission.  
  
a. H. Zhu, T. Liu, D. Zhu, and H. Li, "Robust and Simple N-Party Entangled Authentication Cloud Storage Protocol Based on Secret Sharing Scheme," Journal of Information Hiding and Multimedia Signal Processing, vol. 4, no. 2, pp. 110-118, Apr. 2013.  
  
b. H. C. Huang, J. S. Pan, Y. H. Huang, F. H. Wang, and K. C. Huang, "Progressive Watermarking Techniques Using Genetic Algorithms," Circuits, Systems, and Signal Processing, vol. 26, no. 5, pp. 671-687, Oct. 2007.  
  
c. V. Korzhik, G. Morales-Luna, and K. Nebaeva, "The Capacity of a Stegosystem for the Noisy Attack Channel," Journal of Information Hiding and Multimedia Signal Processing, vol. 3, no. 2, pp. 205-211, Apr. 2012.  
  
d. Y. H. Chen and H. C. Huang, "Coevolutionary Genetic Watermarking for Owner Identification," Neural Computing and Applications, 2014. (DOI:10.1007/s00521-014-1615-z)  
  
Response 18: The above papers have been discussed in the related work section (see section 2.1.4). Specifically, the following text has been added.

“From the perspective of application of SSS scheme, Zhu [23] developed a novel cloud storage protocol based on Shamir’s Secret Sharing scheme called Entangled Authenticated Cloud Storage (EACS), to provide the same quality service for each client in the cloud system and to ensure authentication and integrity in the storage system. The entangled method was used to design a scheme which has four key policies to describe the way that the clients worked with the storage protocol and how the storage cloud treated the clients in the system when one client deleted, removed or modified the files in the system. In other work, Korzhik [24] discussed and analyzed the capacity of a stegosystem for the noisy attack channel by considering the scenario where an attacker may know the cover message of the transmitted stegosignal which was received over a noisy channel. Moreover, after considering a strong upper bound for the SG and a lower union bound for the probability of block error, the author proved that the capacity of a stegosystem for the noisy attack channel is equal to zero. Also, Chen and Huang [25] proposed a co-evolutionary genetic watermarking scheme based on wavelet packet transform to improve the capability to resist specific image processing methods while keeping the quality of the watermarked image acceptable. After testing and evaluating the method, the authors noticed that this characteristic would make genetic watermarking schemes more applicable in real-world applications.”

**Reviewer #4:**

Comment 19: It is not new, only used the algorithm that already used in the previous works. 2. It is difficult to read without examples

Response 19: Thank you for your comment. We clarify the novelty of our method as follows.

The proposed method is novel in that all the previous methods discussed how to send the shares over a secured channel but few researchers discussed how to send over unsecured channels. Those who suggested sending the image over unsecured channels used the encryption process on the whole image. In our proposed method, instead of using encryption on the whole image, only part of the image will be encrypted and at the same time the other information (pixels) of other parts will be changed using the first part. This procedure is applied using the linear independence relationship between the parts. Moreover, this is the first research (known to the authors) that uses the Linear independence relationship between the transmitted shares to hide the information of the generated shares.

To address this concern, we have revised the second paragraph in the introduction section, as follows:

“Many researchers [1-15] discussed methods for sharing an image or data among different users. These methods assume that an attacker cannot gather *t* or more shares. This assumption may not hold true especially in unsecured networks such as a wireless camera network. We found that there is little research focusing on sending shares over such unsecured networks in a trusted and safe way, such that even if an attacker finds *t* or more shares, s/he should not be able to obtain any information about the secret data. The main focus of this paper is to deal with security issues in such scenarios.”

Comment 20: The proposed method insisted that it is fast, but there is no experimental results to display the speed.

Response 20: We have provided these results in Table 2 to Table 4.

Comment 21: What is important things to unsecured channels? It is already secure by using secret image sharing. So, it is important to send the meaning shared images to the receiver not to detect.

Response 21: We appreciate the comment and agree that secret image sharing methods provides security as long as the attacker is unable to obtain a certain number shares. However, if the attacker sniffed the required number of shares (*t* <= *k*, in our case), then she can reconstruct the secret data. The proposed method encrypts the shares and therefore it protects the secret data even if the attacker gets *t* or more shares. We have further clarified it in the paper by revising the second paragraph in the introduction section (please return to Response 19).

Comment 22: How do the proposed method the pixel value 251~255?

Response 22: We appreciate the comment, however we would like to clarify that using “mod 251” is a standard practice in SIS methods. Furthermore, the number of pixels that are greater than 250 in standard images is small, so truncating the pixels that are greater than 250 creates no noticeable effect.

We added the following text in subsection 3.1.1 (Phase 1 - creating parts)

“Using “mod 251” is a standard practice in SIS methods. Furthermore, the number of pixels that are greater than 250 in standard images is small, so truncating the pixels that are greater than 250 creates no noticeable effect.”

Comment 23: In section 4.5.3, the histogram are different with original, that means the algorithm is not good and easy to detect by the histogram attack, but the author insisted that it is strong compared with other methods.

Response 23: Figure 13 describes the histogram of the original images (on left) and the histogram of the shares after applying the proposed method (on right). Since the histogram of the shares after applying the proposed method (on right) is uniformly distributed, it does not provide any information about the secret image. Therefore, we deduce the algorithm is secure against the histogram attack.

We apologize for the ambiguity in section 4.5.3 of the previous version of the paper (now it is section 4.6.3). Accordingly, we have revised the text in section 4.6.3 as follows:

“An ideal encryption algorithm should provide a uniformly distributed image histogram, rendering the histogram useless to attackers. In our tests, Figure 13 describes the histogram of the original images the Lena, Cameraman and Peppers images (on left) and the histogram of their shares after applying the proposed method (on right). Since the histogram after applying the proposed method (on right) is uniformly distributed, it does not provide any information about the secret image. Therefore, we deduce the algorithm is secure against the histogram attack.”

Comment 24: The author have to give the reason why they use "unsecured channels" in the title.

Response 24: As the main objective of this work is to protect the image shares when they are sent over the unsecured channels, we feel it was important to include “unsecured channels” in the title. Please return to the Comment 21 for further explanation. However, we have revised the title to “Secure image sharing method over unsecured channels”, as per the suggestion by Reviewer #6 (via Comment 25).

**Reviewer #6:**

Comment 25: The title must be changed into "Secure image sharing **method** over unsecured channels".

Response 25: Thank you for your suggestion. We changed the title of the paper accordingly.

Comment 26: Authors should present the proposed algorithm for both construction (Fig. 2) and reconstruction phase (Fig.3) in a understandable way

Response 26: Thank you for the comment. The construction and reconstruction algorithms are shown as a flowchart with the steps written in the text and as algorithms. The algorithms have been slightly revised for clarity.

Comment 27: There is no novelty in this work they should avoid using the (new) word because they just used some existing method to enhance the efficiency of the proposed method.

Response 27: We would like to reiterate that the proposed method is novel in that all the previous methods discussed how to send the shares over a secured channel but few researchers discussed how to send over unsecured channels. Those who suggested sending the image over unsecured channels used the encryption process on the whole image. In our proposed method, instead of using encryption on the whole image, only part of the image will be encrypted and at the same time the other information (pixels) of other parts will be changed using the first part. This procedure is applied using the linear independence relationship between the parts. Moreover, this is the first research (known to the authors) that uses the Linear independence relationship between the transmitted shares to hide the information of the generated shares.

To address this concern, we have revised the second paragraph in the introduction section, as follows:

“Many researchers [1-15] discussed methods for sharing an image or data among different users. These methods assume that an attacker cannot gather *t* or more shares. This assumption may not hold true especially in unsecured networks such as a wireless camera network. We found that there is little research focusing on sending shares over such unsecured networks in a trusted and safe way, such that even if an attacker finds *t* or more shares, s/he should not be able to obtain any information about the secret data. The main focus of this paper is to deal with security issues in such scenarios.”

Nevertheless, as suggested, we have tried to avoid the use of “new” in the paper.

Comment 28: The role of image processing is missing. Authors should give the details of reducing the size of image.

Response 28: The focus of the paper was not to present a new image processing method, but to protect the image shares created by the SSS scheme. In our method, the size of the image shares is reduced by a variant of the SSS scheme (specifically using Eq. 5 and Eq. 6).

Comment 29: Authors should give more arguments about the impact of using encryption for a part of image before sharing through unsecured channels.

Response 29: Thank you for the comment. We agree that the corresponding text in the previous version was ambiguous. Therefore we have revised it (in section 5.1) as follows:

“The encryption is used for two main reasons: to generate an encrypted share and to encrypt the initial conditions of the chaotic model. The encryption in the two cases does not have any significant impact on the computation time of the algorithm since only part of the image is encrypted. Furthermore, this impact can be minimized and controlled depending on the size of the encrypted share, which depends on the size of the original image and the number of shares. While the encryption step does not add much to the computational complexity, it is very useful in providing higher security, which is analyzed as follows.”

We hope that the above modification will satisfy the reviewer.

Comment 30: The robustness of the proposed method is not tested against some types of attack in a real time application.

Response 30: We have analyzed the proposed method using the standard metrics such as information entropy analysis, histogram analysis, correlation analysis, plaintext sensitivity analysis and keys space analysis (in sections 4.6.1 to 4.6.5). Besides that, we discussed the performance of the proposed method in sections 5.1-5.4 and the results show that the proposed method is robust and strong against different types of attacks. Please note that section 5.3 and 5.4 have been added in the revised version.

Comment 31: The optimal test in subsection 4.4.3 is not clear how they found out the optimal value of t\_1 between 2 and 20? And what t1 value is more optimal?

Response 31: By using the figure analysis as shown in Fig.12, the last value before increasing is equal to 4. The optimal value is calculated depending on the execution time of the algorithm.

Comment 32: The reference list is very poor. They should put more recent research paper in this area.

Response 32: We have added the following references to the list.

H. Zhu, T. Liu, D. Zhu, and H. Li, "Robust and Simple N-Party Entangled Authentication Cloud Storage Protocol Based on Secret Sharing Scheme," Journal of Information Hiding and Multimedia Signal Processing, vol. 4, no. 2, pp. 110-118, Apr. 2013.

V. Korzhik, G. Morales-Luna, and K. Nebaeva, "The Capacity of a Stegosystem for the Noisy Attack Channel," Journal of Information Hiding and Multimedia Signal Processing, vol. 3, no. 2, pp. 205-211, Apr. 2012.

Y. H. Chen and H. C. Huang, "Coevolutionary Genetic Watermarking for Owner Identification," Neural Computing and Applications, 2014. (DOI:10.1007/s00521-014-1615-z)

**Reviewer #8:**

Comment 33: 1. The paper is on image sharing. Its contribution seems to be the smaller size of the shared images and therefore the faster processing.

Response 33: Thank you for your review and constructive comments.

Comment 34: Section 1: "The share images can be encrypted using any strong encryption algorithm such as Advanced Encryption Standard (AES), but it is computationally expensive, especially in a wireless network scenario where the computation power of camera nodes is limited." In my view, if a camera is used to take and transmit secure images for important purpose, the CPU power should not be a problem. In terms of image transmission, which takes more time, compression or encryption? Of course, this doesn't devalue your work.

Response 34: Thank you very much for your explanation. By providing an example of a camera network scenario, we have tried to show the situations where unsecured channels can lead to loss of confidentiality of secret data (when existing secret image sharing methods are used). Furthermore, the proposed method is not affected by the CPU power, though we have shown that the proposed method can encrypt the images in reasonable time.

Comment 35: In Section 2.2, past works are described. In table 1, some methods in some papers are listed and compared. However these paper mentioned in Section 2.2 are not listed in table 1. Is there difference between the proposed and method in paper [2]?

Response 35: Thank you for the comment. Accordingly, the table was removed and the following text is added in Section 2.2:

“Yuan and Xu [24, 25] proposed to encrypt the image using a symmetric encryption scheme such as the chaos-based technique or one-time pad and distribute the encrypted image with the key to different participants. However, the shares should be the same size as the original image. Therefore, this approach requires a great amount of time to achieve effective encryption. Shi [26] proposed to encrypt the secret image by generating public and private images, then distributing the public image to the participants with the private image which used SSS. However, the problem of increasing the share size between the participants will arise, which will decrease the speed and require a great amount of time to encrypt and decrypt the images. Lang and Malik’s [4,27] method was to encrypt the image using any encryption scheme then distribute the encrypted image to all the participants without exchanging the key by using a three-pass protocol or with a keyless approach. This minimizes the information shared between the dealer and the participants. However, the information can be leaked as decryption is possible for the chain of users. Feldman [28] proposed to share the image using SSS then encrypt each share before distributing them among the participants using a public key. Still, all the data must be encrypted before sending the shares over unsecure channels, which will be time-consuming in encrypting and decrypting.

In contrary to the above works, the proposed method encrypts only a part of the image depending on the number of shares and then shares the images with the encrypted keys. This minimizes the size of the shares and therefore reduces the execution time for sharing and retrieving the image, while achieving security on unsecured channels.”

Comment 36: pp.6, section 3.1.3, "The generated parts are used in equation Eq. 2 used in SSS to create shares.”?

Response 36:We appreciate the comment. We removed “used in SSS” because it was a duplicate. To clarify this point, we have added the following text at the beginning of subsection 4.5 (Number of Shares)

“The part is the image share after applying SSS in phase.1, where the share is the image share after applying SSS in phase 3 as shown in Figure 3. Also, t1 and t2 are parameters used to denote the number of parts and shares, respectively. The parts are used to create a linear independence relationship with other parts. In order to increase the randomness of the proposed method, the parts will then be combined to create a new image. Phase 3 uses the new image as input to create shares using SSS. Therefore, the relationship between two parameters is that increasing the number of parts increases the randomization and enhances the shares in phase 3.”

Comment 37: There are some small editorial deficiencies in the paper, which can be eliminated by a thorough reading.

Response 37: We have tried to edit the noticeable errors in the text.

Comment 38:  A number of experiments have been given to demonstrate the proposed algorithm can work. Execution times are also given. However, it is difficult to see whether the proposed algorithm can work better than others. The proposed algorithm is supposed to be faster than others.

Response 38: Thank you for the comment. We would like to clarify that we tested the proposed method by using the size tests and the speed tests in section 4.4 and we used a standard metrics in 4.5 to compare the proposed method with the standard values. Specifically, the proposed method with different parameters was compared with AES (one round and 9 round version); the results (as can be seen in Table 3) show that the proposed method is faster compared AES. In addition, the benefits of the proposed method are summarized in section 5.2.