Verifiable Secret Sharing Scheme on Images using Watermarking

Chitradeep Dutta Roy [1], Neel Choudhury [1], Amrik Chatterjee [2]and Prof (Dr.) Avishek Adhikari [3]

Department of Computer Science and Engineering , Institute of Engineering and Management [1]
Department of Information Technology , Institute of Engineering and Management [2]
Department of Pure Mathematics, University of Calcutta [3]

Abstract
In this paper, we have implemented Adi Shamir's (k, n) threshold secret sharing scheme onto 24-bit bitmap images. Along with this we have used a concept of watermarking, hash function and a multilayered threshold secret sharing approach to verify the authenticity of each image share before processing instead of decrypting blindly. Moreover this verification scheme allows us to identify dishonest participants who might have gained unauthorized access on secret shares. Consequently, this proposed scheme offers a highly secure mechanism of image secret sharing, which is different from existing traditional methods. Although we have implemented this scheme on images it can easily be used on any digital format as we are dealing with streams of bits.

Keywords:
Secret Sharing, Lagrangian Interpolation, Watermarking, Share Authentication, Hash function, Symmetric key encryption.

1. Introduction

Due to the fast growth of internet and huge data sharing various cryptographic techniques are gaining lots of attention. Many cryptographic techniques are used for secure data transfer between client and server. However in today's scenario we also need to consider situations where the secret is shared by a group of entity. Special interest should be given to situations where a group of individual shares a secret but no one can be trusted fully. Consider a bank vault that can only be opened by a combination code s. The bank has n vice presidents and we want the vice presidents to be able to open the vault if necessary. However we cannot entrust any individual vice president to have the entire right to open the vault. In such scenario it might be necessary to deploy a scheme so that any k vice presidents together can open the vault. This problem can be tackled by a cryptographic method known as Secret Sharing. The concept of (k, n) threshold secret sharing can be used to solve this problem. In this scheme we divide a secret S into n shares and distribute them among n participants. But when we collect shares from k or more participants we can easily reconstruct the secret whereas with k-1 or less shares we cannot find any information about it. Among numerous cryptographic solutions available, secret sharing schemes have been found sufficiently secure in situation where the data is shared by a group of entity.

Now our approach had been in extending this existing secret sharing scheme on data to any 24-bit bitmap (.BMP) color image because the secret mentioned in the above example has fair chance to be an image. In this pursuit we have developed a scheme where we considered each pixel of an image as a data element having any value within a certain range $[0\text{-}2^{24})$ and the entire image as a simple collection of such data elements. After conceptualizing an image in such a fashion we have implemented the (k, n) secret sharing strategy on it to divide the actual image into n shares out of which any given k or more shares can retrieve the original image.

Apart from this in a group-oriented context [5], it is also important to consider the case when some participants submit invalid shares to the combiner either to get some information about the original image or to prevent original image from being reconstructed by the combiner. To prevent such unauthorized attempt to discover the secret we have incorporated the concept of watermarking and hash function onto our (k, n) threshold scheme. This has led us to the generation of a new Verifiable Secret Sharing (VSS) Scheme by which we are able to verify the authenticity of each share before progressing to the part of reconstructing the original image.

2. Background and Related Work

Secret sharing has been a topic of great interest in the field of cryptography. Its importance has increased in recent times as group oriented cryptography is gaining popularity. The first idea of secret sharing was devised independently by Adi Shamir [1] and George Blakley [2] in the year 1979.

Adi Shamir's [1] scheme relies on the idea that one can fit a unique polynomial of degree (k-1) to any set of k points that lie on the polynomial. The method is to create a polynomial of degree (k-1) with the secret as the first coefficient and the remaining coefficients picked at random. Next find *n* points on the curve and give one to

Manuscript received February 5, 2010
Manuscript revised February 20, 2010

each of the participants. When at least k out of the n participants reveal their points, there is sufficient information to fit an $(k-1)^{th}$ degree polynomial to them, the first coefficient of the polynomial is the secret. He used Lagrange's Interpolation to solve this polynomial.

In contrast to Shamir, Blakley [2] specifies the secret as a point in k-dimensional space, and gives out shares that correspond to n hyper planes that intersect at the secret point. Any k such hyperplanes will specify the point, while fewer than k hyperplanes will leave at least one degree of freedom, and thus leave the point unspecified.

There are two other secret sharing schemes that make use of the Chinese Remainder Theorem, namely Mignotte's [4] and Asmuth-Bloom's [3] Schemes. They are threshold secret sharing schemes, in which the shares are generated by reduction modulo the integers (m_i), and the secret is recovered by essentially solving the system of congruence using the Chinese Remainder Theorem.

Our scheme is based on Adi Shamir's Secret Sharing Scheme where a secret ***S*** is distributed among ***n*** participants and can be recovered from any ***k*** or more shares.

To do this we first choose a Prime Number ***p***. Each participant ***i*** is given a unique identifier,

$$ID_t \in Z_p \text{ for } 1 \leq i \leq n$$

where Z_p is the prime field for prime number p

Now we form the standard Shamir's polynomial of (k-1) degree in Z_p

$$F(x) = \left(S + \sum_{j=1}^{k-1} A_j * x^j \right) mod\ p \qquad (1)$$

where ***S*** is the secret and A_j s are random numbers selected from Z_p. Each participant i is given $y_i = F(\mathbf{ID_i})$ as the share.

For the reconstruction of the secret we need shares of any k out of n participants. We use Lagrangian Interpolation over Z_p. When we have any k y_is, we can obtain the secret using the following mathematical expression

$$S = \sum_{j=1}^{k} \left(y_{t_j} * \prod_{1 \leq m \leq k, m \neq j} \frac{ID_{t_m}}{ID_{t_m} - ID_{t_j}} \right) mod\ p \qquad (2)$$

Here in the above equation denotes the y_i or $F(ID_i)$ of the j^{th} participant (considering that his id is ID_i).

Li Bai [6] recently proposed the implementation of (k, n) threshold scheme on the domain of images using bitwise logical operations. Concept of verifiable secret sharing scheme was first envisioned by B. Chor, S. Goldwasser, S. Micali, and B. Awerbuch [7]. They used the concept of cyclic group to achieve Verifiability. Recently in 2007 Rong Zhaoa, Jian-jie Zhaob, Fang Daia and Feng-qun Zhaoa [8] also worked on a secret sharing scheme where they had methods to identify cheaters.

In our work we have extended secret sharing onto images although it is easily possible to implement the scheme on any form of data (e.g. text audio) because it deals with raw bits. And we have achieved verifiability using the concept of watermarking and Hash function. In our method we are embedding a fragile watermark into each share. Before decrypting the secret the hash function is used to check the validity of watermark and thereby authenticity of each share.

Watermarking is a concept of embedding an authentication code or data into a digital medium like image, text, audio or video. Here in our method we are using watermarking to embed a secret key to ensure authenticity of each share before processing. In LSB watermarking on images firstly the authentication code is converted into a bit string and then each bit of it is embedded into the LSB of some specific pixels of the original file. These pixels are chosen randomly depending on the output of some random number generator which is decided by a watermarking key.

3. Methodology

A. Public Entities

To implement our scheme we have made the following entities public.

- A standard Symmetric Key encryption and decryption algorithm. We have used ***Blowfish*** [9] for implementation
- A standard one-way Hash function. We have used ***md5*** [10] for the implementation.
- A standard Random Number Generation Algorithm that generates a random sequence depending on a seed.

B. Encryption and Distribution

Step 1: The secret image is encrypted by the standard encryption algorithm using an encryption key E

mentioned above. The hash value of the key E is made public.

Step 2: The encrypted image is divided into n shares such that any k shares can be used to retrieve the encrypted image byte by byte using the eqt[n] (1) stated above. While using our method stated in eqt[n] (1) for image we are considering an image as a stream of 1 byte (1/3[rd] of each 24-bit pixel). The prime number p for this scheme is 7996369. It is chosen because it is largest prime below 2^{23}. For that reason all ID_is and A_js are chosen below 7996369. As a result $F(ID_i)$ s for each participant i is below 2^{23} and is stored in the form a pixel (using only its most significant 23 bits) in each share image for the respective participant. Because of this method each byte of the secret image is transformed into a pixel of each share image so the size of a share image is 3 times the size of the secret image.

Step 3: We now randomize the last bits of the share pixels with 0 and 1s to obfuscate or to disguise the watermark so that it becomes difficult to determine the presence of a watermark using standard cryptanalysis.

Step 4: The encryption key E is then embedded using LSB watermarking into the last bits of some specific pixels of each share decided by the watermarking key W using the random number generation algorithm mentioned above. Thus E is used for encryption of the image as well as a code to check the genuineness of shares. E is used for watermarking because after distribution dealer is not supposed to have any contact with shareholders so in that case the secret encryption key will be sent in each share as a hidden watermark. In this way dealer can maintain secrecy too because he doesn't need to convey the key directly to the shareholders via some insecure channel.

Step 5: Now the watermarking key W is divided into n pieces using (k, n) threshold scheme described in eqt[n] (1) so that k out of n shares can reveal the watermarking key. Now each such piece is kept in the first pixel of each image share. It can also be kept at some other specific pixel of the share images.

C. Decryption and Reconstruction

Step 1: We take any k out of n image share as input.

Step 2: We extract the first pixel from each of this k shares and using the Lagrangian Interpolation method described in eqt[n] (2) we get the watermarking key W.

Step 3: Using W we determine the locations of the pixels which contain the segments of the encryption key E in their last bits. We then rejoin those bits to form E from each share.

Step 4: Now the hash value of E retrieved from each share is checked with the public hash value of E produced in step 1 of encryption for consistency. From their inconsistency we can determine that some participants or shareholders incidentally or accidentally have given wrong shares. If there is inconsistency with hash values of any E we discontinue the process and mark the dishonest shareholders. Otherwise we proceed with no interruption.

Step 5: Now we read each pixel from all the k shares and using the Lagrangian Interpolation described in eqt[n] (2) with p = 7996369 we get an encrypted stream of bytes and all these bytes combined produces the encrypted pixel set of the original image

Step 6: Using encryption key E obtained from step 4 we decrypt the encrypted image using the standard scheme which is used for encryption. And finally the original imaged is obtained with absolutely no loss.

4.Merits and Extensibility

In our (k, n) threshold secret sharing scheme the time complexity during the distribution of shares according to the eqt[n] (1) is O(pnk) where p is the number of pixels in the secret image. And during the reconstruction the time complexity is of the order $O(pk^2)$ according to the eqt[n] (2).

As we are transforming each byte from each pixel of the secret image into one pixel of the share image so the size of the share image increases by 3 fold of the original secret image. The dimension would be approximately √3 times of the secret image.

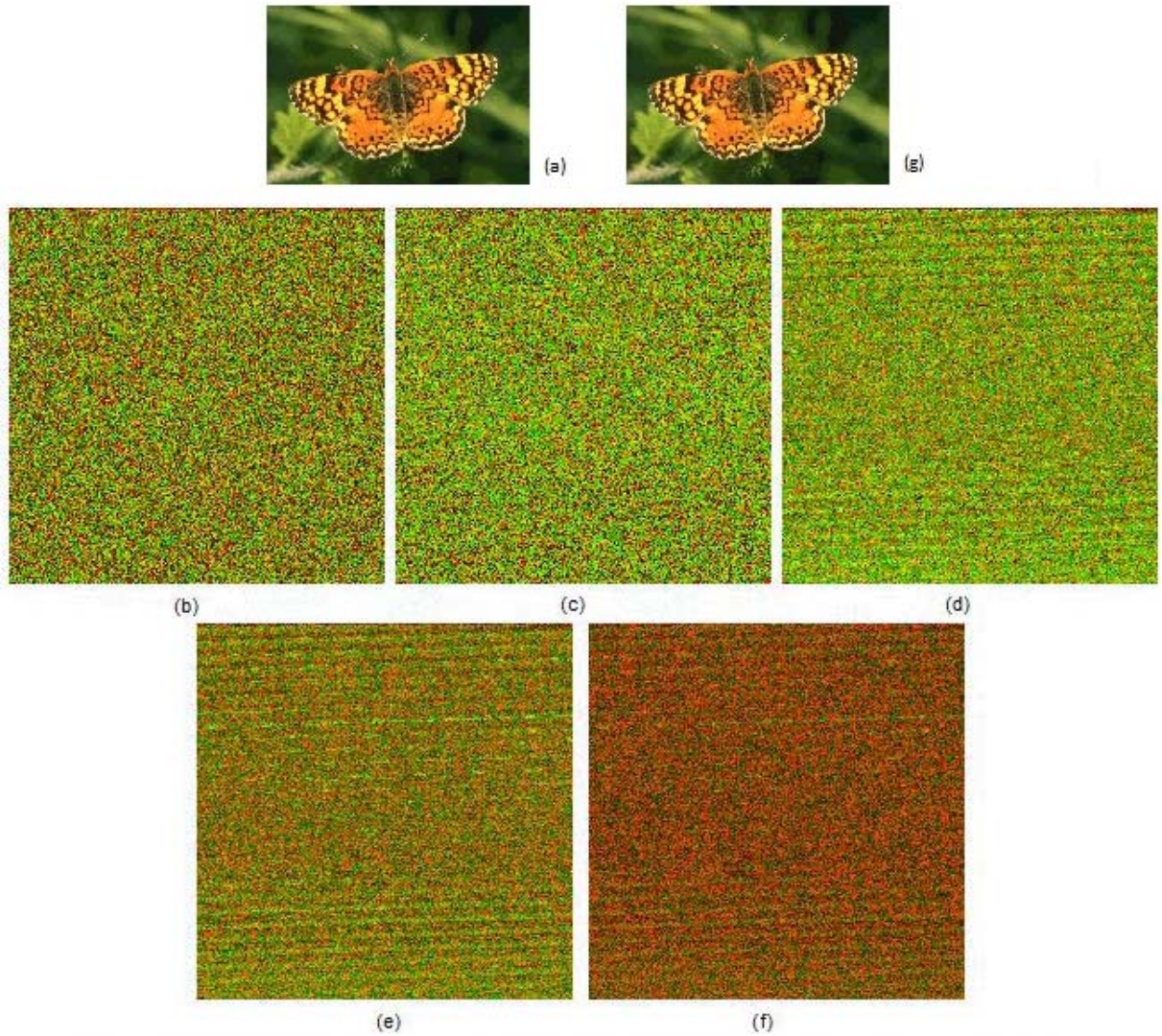
In our Verifiable Secret Sharing Scheme we are taking a multilayered approach of (k, n) threshold scheme where we not only divide the image data into n shares but also the watermarking key W to ensure higher security. And our verification system based on hash function also enables us to point out exactly who is the dishonest participant without revealing any secret information.

For embedding the secret verification key we are using LSB watermarking in Spatial Domain because it is the most fragile watermarking scheme. So the slightest modification by any cheater can be easily detected which increases our security.

Very importantly as this scheme deals with bits not some specific structures so it is easily extensible to any formats like JPEG, PNG etc. Moreover the concept can be extended to other digital format besides image.

5. Application

Our scheme is ideally suited for organizations in which a group of mutually suspicious individuals with conflicting interests must cooperate for a piece of secret. So this scheme will have opportunities in military organizations, financial groups, and research labs having many heads or authorities. These organizations can share an important piece of image like a satellite image, blueprint of an experiment or a scanned copy of some important document without giving sole responsibility to any particular individual. This system ensures that the secret remains secure but always available because despite some unwilling participants the secret can be deciphered from k others of the rest while a majority of unwilling participants can also block the secret. With the progress of distributed computing and group cryptography this type of security system is certainly going to prevail in this form or some modified one as they are less vulnerable.



(a) Original image (147x100),(b) Share 1,(c) Share 2,(d) Share 3,(e) Share 4,(f) Share 5 (each 210x210)
(g) Reconstructed image from share (1,3,5) with absolutley no loss in resolution and clarity (147x100)

Figure 1

6. Experimental Results and Illustrations

A sample experiment with an image is shown in Figure1.We have a taken a 24-bit 147 x 100 colored bitmap image (Figure 1(a)). Then we have divided the image into 5 shares such that form any 3 of the 5 shares the original image can be retrieved. Each share is having a resolution of 210 x 210 and all are distinctly different form one another and obviously from the original secret image (Figure 1(b)-(f)). And in this experiment no share image reveals any information about the secret image that is being divided. Now we have taken all possible combination of three distinct shares and after decryption we got back the original image. For example in the illustration showed in Figure 1(g) we showed share 1, 3 and 5 being used. However with less than 3 shares we could not get any information about the original image.

And many such experiments are done on images of different resolutions and taking different values of k and n as well. All were successful in retrieving the secret image.

In order to check the failsafe nature of the system in presence of dishonest participants, share 1 and share 3 (Figure1 (b), (d)) was modified. As a result the watermark embedded in the shares became corrupt. When share 1, 3 and 5 were used for regeneration of the secret the system successfully identified the share 1 and 3 as invalid checking the hash values with the public hash value of watermark. And so the system stopped reconstruction. All experimental results validated the fact that the proposed scheme produced expected outcome in distribution, reconstruction and authentication as well.

7. Conclusion

This paper presented a novel approach of a verifiable secret sharing scheme using the concept of watermarking and hash function. Although we have used a 24-bit bitmap image for case study we can implement this technique on any digital format such as text, audio.

The complete process was demonstrated with the help of examples and it was found that proposed approach performed well in every case.

References

[1] Adi Shamir, How to share a secret, Communications of the ACM, v.22 n.11, p.612-613, Nov. 1979

[2] G.R. Blakley, Safeguarding cryptographic keys. In Proceedings of the National Computer Conference 1979, volume 48 of AFIPS Conference Proceedings, pages 313–317, 1979.

[3] C. A. Asmuth and J. Bloom, A modular approach to key safeguarding. IEEE Transactions on Information Theory, IT-29(2):208–210, 1983.K. Elissa, “Title of paper if known,” unpublished.

[4] M. Mignotte, How to share a secret. In T. Beth, editor, Cryptography-Proceedings of the Workshop on Cryptography, Burg Feuerstein, 1982, volume 149 of Lecture Notes in Computer Science, pages 371–375. Springer-Verlag, 1983.

[5] C. Tartary and H. Wang, “Dynamic Threshold and Cheater Resistance for Shamir Secret Sharing Scheme” Lecture Notes in Computer Science, vol. 4318, pp 103 - 117. Springer-Verlag.

[6] Li Bai, "A Reliable (k, n) Image Secret Sharing Scheme," dasc, pp 31-36, 2nd IEEE International Symposium on Dependable, Autonomic and Secure Computing (DASC'06), 2006.

[7] B. Chor, S. Goldwasser, S. Micali and B. Awerbuch, Verifiable secret sharing and achieving simultaneity in the presence of faults. In Proc. 26th IEEE Symposium on Foundations of Computer Science (FOCS ’85), pages 383–395, IEEE Computer Society, 1985.

[8] Rong Zhaoa, Jian-jie Zhaob, Fang Daia and Feng-qun Zhaoa, A New Image Secret Sharing Scheme to Identify Cheaters, 2007.

[9] Bruce Schneier, Description of a New Variable-Length Key, 64-bit Block Cipher (Blowfish). FSE 1993, pages 191-204

[10] R. L. Rivest, The MD5 Message-Digest Algorithm, http://tools.ietf.org/html/rfc1321

Chitradeep Dutta Roy is a student pursuing his B.Tech degree (final year) in Computer Science and Engineering from Institute of Engineering and Management, Kolkata, West Bengal, India.

Neel Choudhury is a student pursuing his B.Tech degree (final year) in Computer Science and Engineering from Institute of Engineering and Management, Kolkata, West Bengal, India

Amrik Chatterjee is a student pursuing his B.Tech degree (final year) in Information Technology from Institute of Engineering and Management, Kolkata, West Bengal, India

Dr. Avishek Adhikari did his PhD in Cryptography from Indian Statistical Institute. Presently he is a faculty member of Department of pure mathematics, University of Calcutta, India. He received ISCA Young Scientist Award in Mathematical Science Section (including Statistics) for the year 2007-08. He is the secretary of Institute for Mathematics, Bioinformatics, Information-technology and Computer-science. His current areas of research are Combinatorial Cryptography, Secret Sharing, DNA Cryptography, and Graph Theory.