

RECURSIVE HIDING OF SECRETS IN VISUAL CRYPTOGRAPHY

Meenakshi Gnanaguruparan¹ and Subhash Kak²

ADDRESS: (1) Advanced Micro Devices, 5204 East Ben White Blvd., Austin TX 78741 USA gmeens@hotmail.com and (2) Department of Electrical & Computer Engineering, Louisiana State University, Baton Rouge, LA 70803-5901 USA kak@ee.lsu.edu.

ABSTRACT: This paper introduces the concept of recursive hiding of secrets in visual cryptography. This provides a method of hiding secrets recursively in the shares of threshold schemes, which permits an efficient utilization of the data. We also describe a possible use for authentication.

KEYWORDS: Visual cryptography, threshold schemes, recursive hiding of secrets, authentication.

1 INTRODUCTION

In a (k, n) threshold scheme of protecting a secret [1,2], k out of n pieces of the secret must be brought together before the secret is revealed. In these schemes the pieces are given equal weights, although schemes of asymmetric weights can also be devised [3].

Threshold schemes have found many applications in various types of cryptographic protocols, including secure multiparty computation, escrow key recovery schemes, and electronic cash. A popular embodiment of a threshold scheme is the method of visual cryptography [4, 5], where the secret image is split into two separate random images called *shares*. To decrypt the encrypted information, the shares are stacked one on top of the other, and the secret image appears. These are examples of systems using 2 shares. In more complex versions, the image under consideration may be divided into n shares where k out of these n shares will suffice to retrieve the original image. Each pixel of the image is divided into subpixels and the number of subpixels determines the number of shares.















| Original Pixel | | Share 1 | Share 2 | Result |
|---|-----------|---|---|---|
|  | $p = 0.5$ |  |  |  |
| | $p = 0.5$ |  |  |  |
|  | $p = 0.5$ |  |  |  |
| | $p = 0.5$ |  |  |  |

Figure 1. Basic concept of visual cryptography.

One need not consider the shares in the manner of visual shares. It works as well consider each image as a bit string where the two shares are exclusive-ORed together to provide the original image.

In the example shown in Figure 1, each pixel is divided into a black and a white subpixel placed next to each other. For the case of a white pixel, one of the two combinations of subpixels will be chosen with a probability of 0.5 to represent the pixel in each of the shares. When these shares are placed one on top of the other, the pixels are visually ORed and hence a white pixel looks gray (half black and half white) to the human eye. The pixels are chosen in a similar manner for the case of a black pixel. But when the subpixels are visually ORed, the two black subpixels placed next to each other appear as a single black pixel.

Threshold schemes have the disadvantage that their efficiency is low. In certain applications, such as storing a key amongst n trustees, this is not an issue. But in communication problems, where the shares are transmitted to the destination, the low efficiency may be of concern.

To increase this efficiency, we propose the method of *recursive hiding of secrets*, where the shares themselves are built out of useful information, in a recursive manner. We show that this allows for the transmission of additional information that can be used for authentication or it might serve as a secret channel.

2 RECURSIVE HIDING OF SECRETS

In recursive hiding of secrets, several additional messages can be hidden in one of the shares of the original secret image. The secret images that are to be hidden are taken according to their sizes from the smallest to the largest. The smallest secret image is divided into two shares using the idea of visual cryptography. These shares are placed below or beside each other (concatenated), and they now are taken to constitute the first share of this secret image. The second share is obtained in such a manner that if the two shares are overlaid, the secret image under consideration is revealed.

This process is recursively repeated. The shares of the last secret image constitute the first share of the original secret image and corresponding second share for the original image is obtained. Thus two shares that have a random distribution of black and white pixels are sent to the recipient. One of the shares is sent as the secret key and the other is sent as the cipher.

The algorithm used to embed the secrets in one of the shares is also sent to the recipient as part of the secret key.

It must be noted that the share of the original secret image that contains the recursively-hidden information must contain both the shares of the last hidden secret image. This imposes a constraint on the size of the secret images with respect to the original secret image.

An example to help the reader understand this concept better is shown in Figure 2. The original secret image under consideration is of size 4×4 . The first secret image is of size 2×2 . The two shares of this image are obtained based on the idea of visual cryptography. The second secret image is of size 4×2 . To obtain this second secret image, the shares of the first secret image are placed one below the other to obtain a share of size 4×4 . This forms Share 1 of the second secret image. The second share of the second secret image is obtained such that if the two shares are overlaid, the second secret image is revealed.

Suppose, on the first share, the first subpixel is white and the second subpixel is black, and suppose the first pixel in the second secret image is a black pixel. In order to reveal a black pixel on overlaying the shares, the first and second subpixels in the second share have to be black and white respectively. Now the two shares of the second secret image are concatenated by placing them beside each other to form a single share of size 4×8 . This forms the first share of the original secret image. The second share of the original secret image is now obtained. It is seen that when these two shares are overlaid, the original secret image is revealed. Thus in the first share of the original secret image, a secret

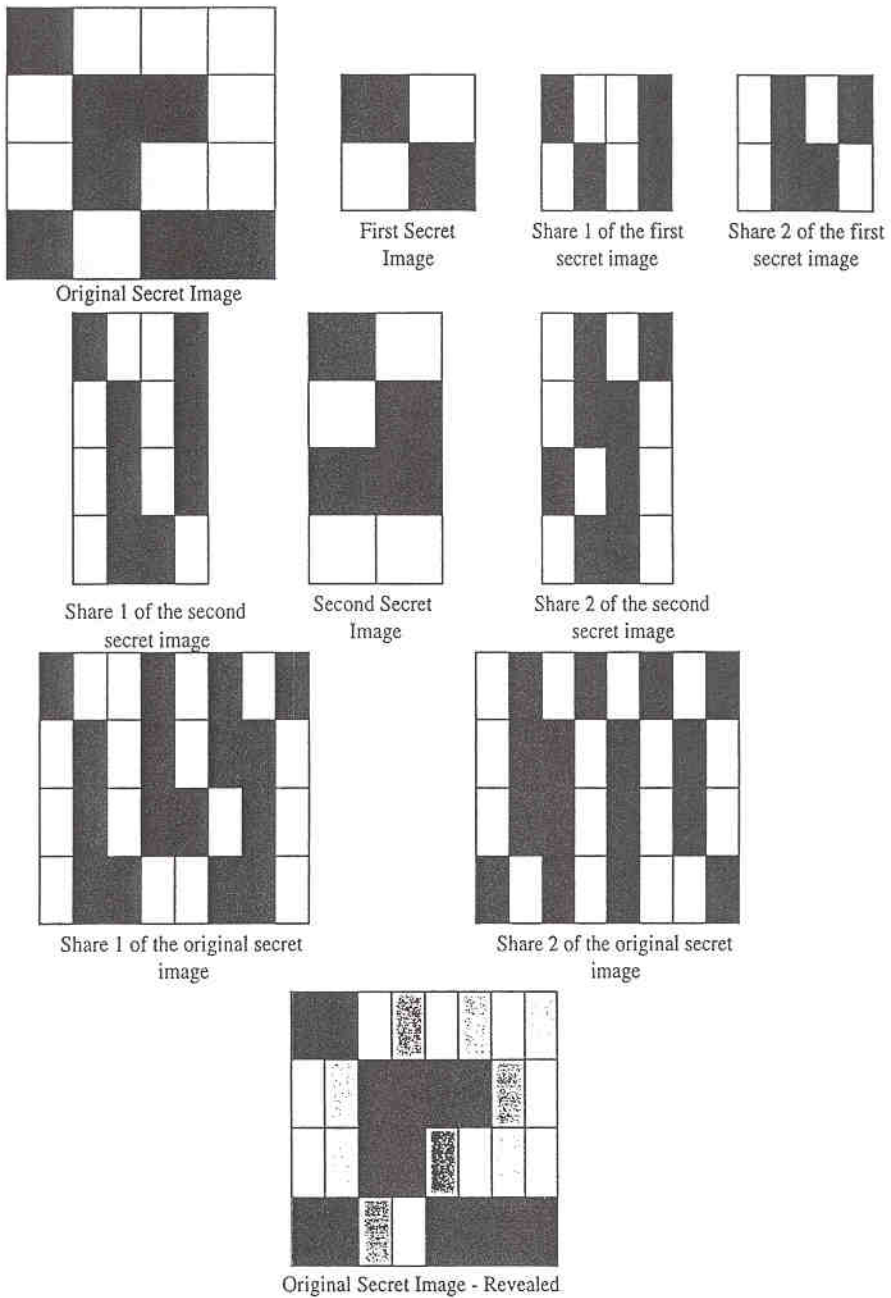


Figure 2. Process of recursive hiding of secrets

image of size 4×2 is hidden and in the first share of this secret image, a secret image of size 2×2 is hidden.

Figure 3 shows an illustration of recursive hiding. The original secret image considered is the *Lena* image of size 256×256 . In the first share of this image, the image of the *Smiley Face* (first secret image), which is of size 128×128 , and the image of *Mona Lisa* (second secret image), which is of size 256×128 , have been hidden recursively.

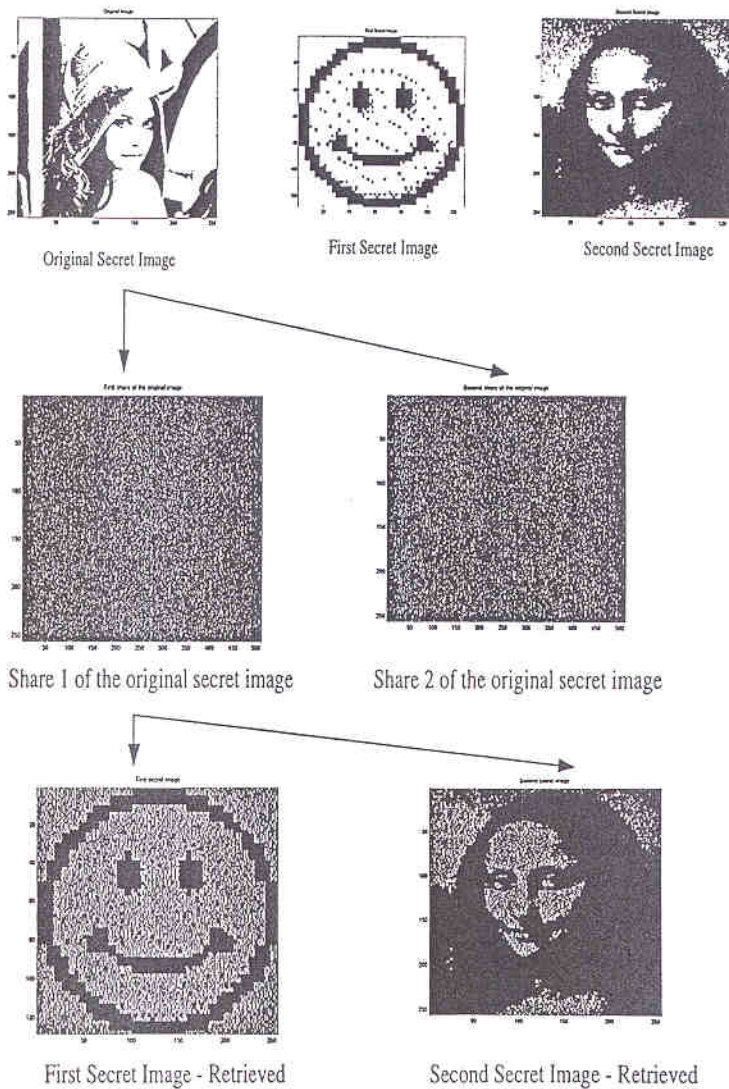


Figure 3. Example of recursive hiding; the two shares hide three images.

Recursive shares

The size of the secret in the recursive hiding of secrets increase by a factor of two as one goes from the smallest to the largest. The smallest secret would be a single bit. At the next level it will be an image of size 2×1 ; the next will be an image is of size 2×2 ; and so on, until the full image size has been reached.

To compute the efficiency in such a system it is convenient to consider recursive secret hiding for one-dimensional messages. Each bit of the message can be mapped into two shares according to the rule:

| Bit | Share 1 / Share 2 |
|-----|-------------------|
| 0 | 1 / 0 or 0 / 1 |
| 1 | 0 / 0 or 1 / 1 |

In other words, '0' maps into different bits, whereas '1' maps into same bits.

Suppose a 16-bit long message M is expressed in shares S_1 and S_2 . Let the secrets be hidden recursively in share S_1 . For example,

M : 0110110101101100
 S_1 : 1110010011101010
 S_2 : 0111011001111001

The share S_1 stores the following secrets in its first 2, 4, 8 bits and in the entire sequence:

| Secret | Share 1 + Share 2 |
|----------|-------------------|
| 1 | 11 |
| 10 | 1110 |
| 0101 | 11100100 |
| 11110001 | 1110010011101010 |

Each of these secrets is obtained by taking the two halves of the sequence on the right and combining the individual bits according to the rule described above.

We see that the 32 bits of S_1 and S_2 allows us to transmit 31 bits of information. These are the $1+2+4+8 = 15$ bits of recursively hidden secrets in S_1 and the 16 bits of S_2 . In other words, the efficiency of this system is near 100%.

In general, if the two shares are of length 2^m each, the total information that can be transmitted will be $2^{m+1} - 1$.

The method of constructing the recursive secrets has assumed that the bits are combined in their natural order. But this need not be the case and the knowledge of the groups that are combined to yield the secrets can be restricted to interested parties.

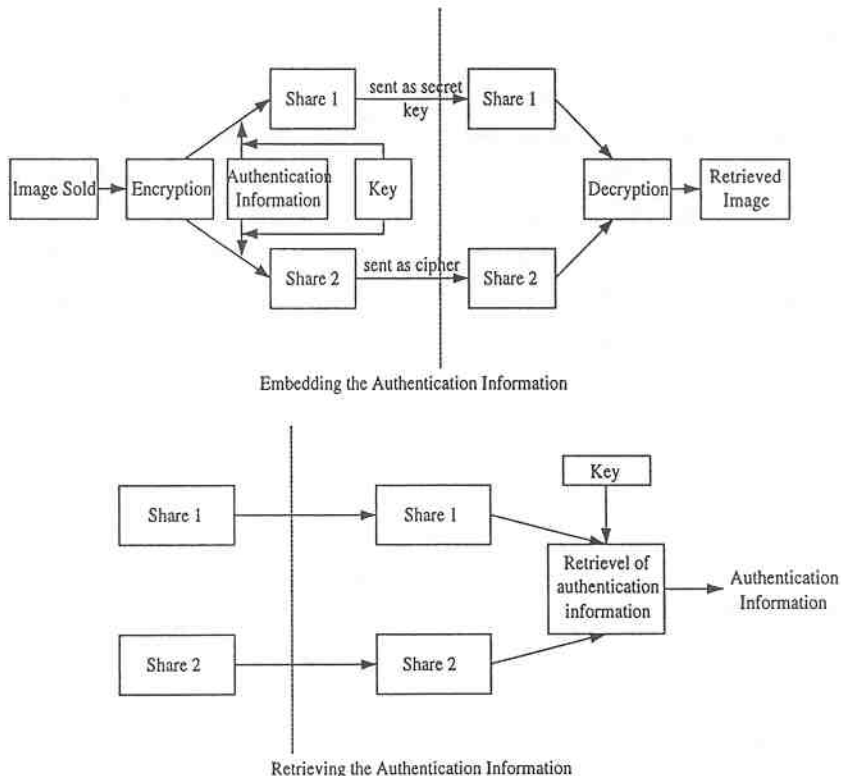


Figure 4. Authentication of images using recursive hiding.

3 AUTHENTICATION OF IMAGES USING RECURSIVE HIDING OF SECRETS

Let us suppose that a museum sells a number of digital images. The images could be sent to the buyers in the form of shares. The museum could embed information specific to the buyer recursively in the shares of the image. Therefore the shares have to be retained in order to prove the authenticity of the image. Say, the museum sells the image to Alice. Alice may post the image obtained by combining the shares in her web site. But the museum may not know that this web site belongs to someone who bought the digital image.

Therefore Alice is required to prove the authenticity of her image by sending

back the shares of the image to the museum.

The museum checks the shares for the owner specific information and thus the image is authenticated. In this case, it is not necessary to reveal the hidden information to the buyer. Figure 4 shows the schematic of a method in which authentication information is embedded and retrieved from the shares of the image that is bought. The authentication information is embedded using another key, which is known only to the seller. When the shares are sent back to the seller, he uses this key to retrieve the authentication information.

4 CONCLUDING REMARKS

The concept of recursive hiding of secrets provides a means to an efficient exploitation of the excess bits used in threshold schemes. We have shown its use for authentication but other applications, such as escrow/key recovery and multiparty communication, can be likewise developed.

Although we described the process mainly in terms of shares, it also works if images are represented as bit-strings and the shares are subsequently combined. We show that our method makes it possible to obtain near 100% efficiency, thus overcoming the main disadvantage of classical visual cryptography.

REFERENCES

1. Blakely, G. R. 1979. Safeguarding Cryptographic Keys. *Proceedings of the National Computer Conference, American Federation of Information Processing Societies Proceedings*. 48: 313-317.
2. Shamir, A. 1979. How to Share a Secret. *Communications of the ACM*. 22: 612-613.
3. Kak, S. 1982. On Asymmetric Secret Sharing. *LSU ECE Technical Report*. May.
4. Naor, M. and A. Shamir. 1995. Visual Cryptography. *Advances in Cryptology- EUROCRYPT*. 950: 1-12.
5. Ateniese, G., C. Blundo, A. De Santis, and D. R. Stinson. 1996. Visual Cryptography for General Access Structures. *Information and Computation*. 129: 86-106.

BIOGRAPHICAL SKETCH

Meenakshi Gnanaguruparan obtained her Bachelors degree in Electronics and Communication Engineering from the University of Madras, Chennai, India, and her MS in Electrical Engineering from Louisiana State University in December 2000. She currently works for Advanced Micro Devices, Austin, Texas.

Subhash Kak has been a professor of Electrical and Computer Engineering at Louisiana State University since 1979. His research areas include cryptography, information and computing, He is author of 12 books and numerous research articles.