www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 2 Issue 1 Jan 2013 Page No. 265-303

Secure Visual Cryptography

R Yadagiri Rao

Introduction:

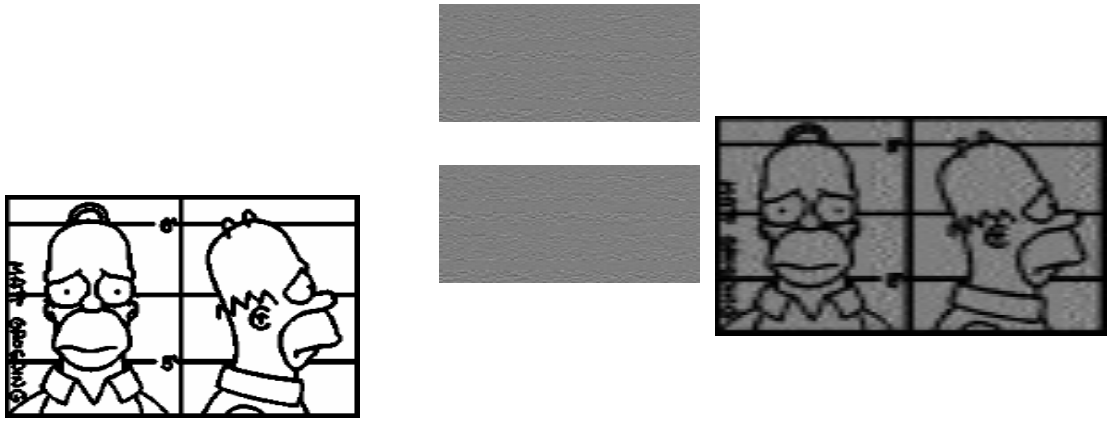
An effective and secure protection of sensitive information is the primary concerned in
Communication systems or network storage systems. Never the less, it is also important for any information process to ensure data is not being tampered with. Encryption methods are one of the popular approaches to ensure the integrity and confidentiality of the protected information. However one of the critical vulnerabilities of encryption techniques is protecting the information from being exposed. To address these reliability problems, especially for large information content items such as secret images (satellite photos or medical images), an image secret sharing schemes (SSS) is a good alternative to remedy these types of vulnerabilities.
With the rapid advancement of network technology, multimedia information is transmitted over the Internet conveniently. While using secret images, security issues should be taken into consideration because hackers may utilize weak link over communication network to steal information that they want. To deal with the security problems of secret images, various image secret sharing schemes have been developed.
Because of the popular usage of images in network application in recent years, the way of sharing secret image has attracted wide attention. Noor and Shamir proposed first the idea of visual cryptography in 1994. The scheme provides an easy and fast decryption process that consists of Xeroxing the shares onto transparencies and then stacking them to reveal the shared image for visual inspection. The scheme which differs from traditional secret sharing does not need complicated cryptographic mechanisms and computations. Instead it can be done directly by the human visual system, without the aid of computers. However the generated noisy share may be suspicious to invaders and their scheme had 2n pixel expansion at best case. Visual cryptography scheme eliminates complex computation problem in decryption process, and the secret images can be restored by stacking operation. This property makes visual cryptography especially useful for the low computation load requirement.
Iwamoto and Yamamoto in 2002, worked on an n-out-of-n visual secret sharing scheme for gray-scale images. They developed a secret sharing scheme that encodes gray-scale images with a limited number of gray levels. The loss in the contrast is so large such that the recovered image is distorted. In other methods that construct a visual secret sharing scheme with a general access structure for plural secret images have been proposed .They have shown that most previous work of visual cryptography scheme for plural image suffered from the leak out of some information in each share about the other secret images of the scheme. The systems suffered from the deterioration of the image quality in addition to the weakness in the security and there are pixels

expansion step in all of method so needed some computation must be applied to reproduce the secret image.

Taking limited bandwidth and storage into consideration two criteria pixel expansion and number of shares encoded is of significance. Smaller pixel expansion results in smaller size of the share. Encoding multiple secret images into the same share images requires less overhead while sharing multiple secrets. Meaningful shares avoid attention of hacker considering the security issues over the communication channels. To meet the demand of today's multimedia information gray and color image format should be encoded by the schemes.

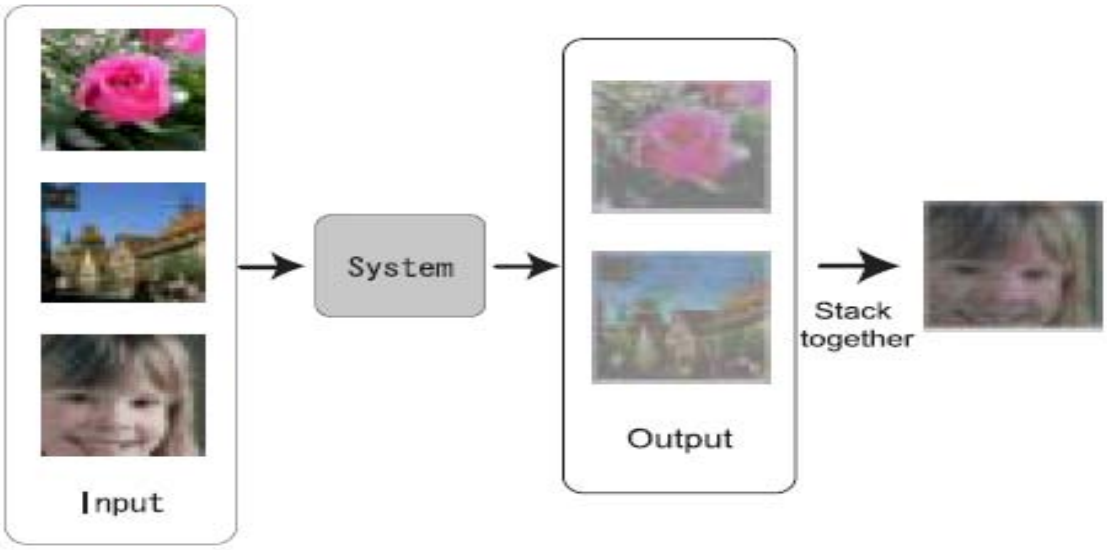
Existing System

In the existing system the dealer or sender takes a secret image and encodes into shares. After encoding this shares are sent to participants. The receiver collects the shares and stack to get decoded secret image. Here no verification is done so easy cheating is done.



Proposed System

In the proposed system the dealer or sender takes one secret image and verification image. These two images are encoded into shares, after encoding sends one secret share and one verification share to the participants. Each participant verifies the share and other participant secret share reveals the secret image. In this way **Cheating is avoided** .



Basic idea of proposed System History:

Wu and Chen were first researchers to present the visual cryptography schemes to share two secret images in two shares. They hidden two secret binary images into two random shares, namely A and B, such that the first secret can be seen by stacking the two shares, denoted by A⊗ B, and the second secret can be obtained by first rotating AΘ anti -clockwise. They designed the rotation angle Θ to be 90◦. However, it is easy to obtain thatΘ can be 180◦ or 270◦.
To overcome the angle restriction of Wu and Chen's scheme, Hsu et al. proposed a scheme to hide two secret images in two rectangular share images with arbitrary rotating angles. Wu and Chang also refined the idea of Wu and Chen by encoding
shares to be circles so that the restrictions to the rotating angles (Θ = 90◦, 180◦ or 270◦) can be removed.
In 1994 the basic problem of visual cryptography was introduced by Noor and Shamir. In visual cryptography we are dealing with the problem of encrypting pictures in a secure way such that the decryption can be done by the human visual system. The encryption of a secret image is achieved by encoding the information into several shadow images, called shares. The decoding is done by printing the shares on transparencies and stacking them. The system can be used by everyone without any knowledge of cryptography and without performing cryptographic computations. This is the major difference to usual cryptography schemes. This is the major difference to usual cryptography schemes where the secret information, represented as numbers, is encrypted by using one-way functions. The decryption can only be done if one knows the appropriate secret key.
Noor and Shamir have described a k out of n (with k<=n) system where the secret is encoded in n shares and the decoding can be done by stacking k or more transparencies. Using less than k transparencies won't reveal the secret not even to an infinitely powerful cryptanalyst.
In general it is possible that k and n are reasonably big numbers. In order to reveal the secret information it is necessary to have at least k people stack their shares together. Since k can be reasonably big, it might be very unlikely to find a coalition of at least k traitors who will be willing to misuse their

shares, in general one share-holder might not even know k-1 other share-holders. Obviously, it is much easier to find 0<t<k traitors who are looking for a way to sabotage the system. We assume that t<k share-holders stack their shares together and publish the information so that other small groups of less than k people can stack their shares on top of the published information and will therefore be able to reveal the actual secret. (It is possible to iterate the scenario in such a way that at first the information of t then t+t' shares etc. gets published.) Since there is no way in keeping t people from publishing the accumulated information we are looking for a possibility to trace the traitors.

S J Shyu et al were first researchers to advise the multiple secrets sharing in visual cryptography. This scheme encodes a set of $n \geq 2$ secrets into two circle shares. The n secrets can be obtained one by one by stacking the first share and the rotated second shares with n different rotation angles. To encode unlimited shapes of image and to remove the limitation of transparencies to be circular, Fang offered reversible visual cryptography scheme. In this scheme two secret images which are encoded into two shares; one secret image appears with just stacking two shares and the other secret image appears with stack two shares after reversing one of them. Jen-Bang Feng et al developed a visual secret sharing scheme for hiding multiple secret images into two shares. The proposed scheme analyzes the secret pixels and the corresponding share blocks to construct a stacking relationship graph, in which the vertices denote the share blocks and the edges denote two blocks stacked together at the desired decryption angle. According to this graph and the pre-defined visual pattern set, two shares are generated.

To provide more randomness for generating the shares Mustafa Ulutas et al advised secret sharing scheme based on the rotation of shares. In this scheme shares are rectangular in shape and are created in a fully random manner. Stacking the two shares reconstructs the first secret. Rotating the first share by 90° counterclockwise and stacking it with the second share reconstructs the second secret.

Tzung-Her Chen et al offered the multiple image encryption schemes by rotating random grids, without any pixel expansion and codebook redesign. A non-expansion reversible visual secret sharing method that does not need to define the lookup table offered by Fang . To encode four secrets into two shares and recovering the reconstructed images without distortions Zhengxin Fu et al intended a rotation visual cryptography scheme. Rotation visual cryptography scheme construction was based on correlative matrices set and random permutation, which can be used to encode four secret images into two shares. Jonathan Weiretal suggested sharing multiple secrets using visual cryptography. A master key is generated for all the secrets; correspondingly, secrets are shared using the master key and multiple shares are obtained.

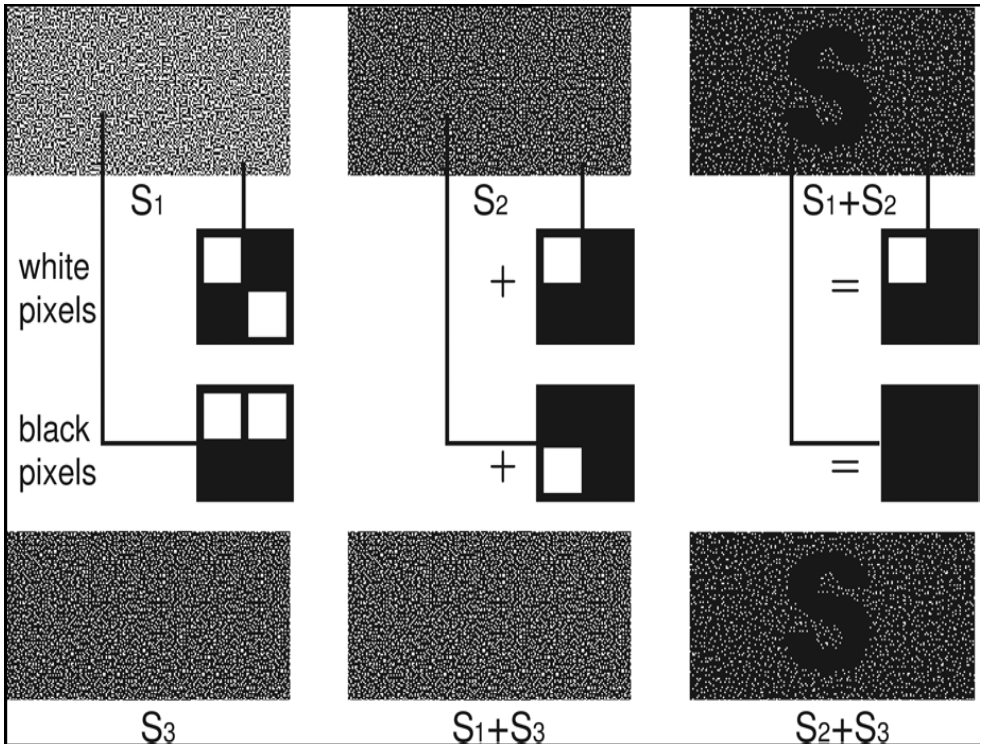
Black and White Images:

Visual cryptographic solutions operate on binary or binarized inputs. Therefore, natural (continuous-tone) images must be first converted into halftone images by using the density of the net dots to simulate the original gray or color levels in the target binary representation. Then, the halftone version of the input image is used instead of the original secret image to produce the shares. The decrypted image is obtained by stacking the shares together. Because binary data can be displayed either as frosted or transparent when printed on transparencies or viewed on the screen, overlapping shares that contain seemingly random information can reveal the secret image without

additional computations or any knowledge of cryptographic keys.
However, due to the nature of the algorithm, the decrypted image is darker, contains a number of visual impairments, and most of visual cryptography solutions increase the spatial resolution of the secret image. In addition, the requirement for inputs of the binary or dithered nature only limits the applicability of visual cryptography.

Most of the existing secret sharing schemes are generalized within the so-called {k,n}-threshold framework that confidentially divides the content of a secret message into n shares in the way that requires the presence of at least k, for $k \leq n$, shares for the secret message reconstruction, Thus, the framework can use any of $n!/(k!(n-k)!)$ possible combinations of k shares to recover the secret message, whereas the use of $k-1$ or less shares should not reveal the secret message.

Use of digital media has exploded in the past few years, primarily due to several distinct advantages that digital media can offer over analog media. These advantages include higher quality, easier editing, perfect copying and easier and more efficient transmission over information network. The wide dissemination of digital media also creates some potential problems. Due to the popularity of Internet commerce and digital library applications, the intellectual property right (IPR) protection is becoming increasingly important. Content providers will be reluctant to provide their valuable contents if they are not assure
d that their contents are securely protected. Some good examples are the deplo
yments of the digital versatile disk (DVD) market and the online music market.



VCS and the structures of sub-pixels

The IPR management and protection issue is currently being addressed in the emerging MPEG-4 standards for moving pictures compression. Several technologies have been developed for IPR protection. One is conditional access through encryption. The digital media will be scrambled before it is distributed. Only authorized users who have the proper key for decryption can access the clear content. The other one is digital watermarking that securely embeds hidden message into the multimedia data to identify the owner, or the buyer of a digital media. These two techniques are complementary to each other. We focus on conditional access through encryption in this project. Digital images/video are often communicated or distributed over non-private channels, such as satellite links, cable television networks, wireless networks, and the Internet.

Conditional access systems for private digital image/video transmission or storage are a necessity for many applications, for example, pay-tv, confidential videoconferences, confidential facsimile transmissions, and medical image transmission and storage in a database. In general, complex cryptography techniques make cracking of the system difficult, but are also expensive to implement. Since digital video transmission system usually includes a compression module that aims to reduce the transmitted bit rate, the cryptography techniques have to be carefully designed to avoid potential adverse impact on the compression efficiency, and on the functionalities that the compression format provides.

VC has been studied intensively since the pioneer work of Noor and Shamir. Most of the previous research work on VC focused on improving two parameters: *pixel expansion* and *contrast*. In these cases, all participants who hold shares are assumed to be semi-honest, that is, they will not present *false* or *fake shares* during the phase of recovering the secret image. Thus, the image shown on the stacking of shares is considered as the *real secret image*.

Nevertheless, cryptography is supposed to guarantee security even under the attack of malicious adversaries who may deviate from the scheme in any way. We have seen that it is possible to cheat in VC, though it seems hard to imagine. For cheating, a cheater presents some fake shares such that the stacking of fake and genuine shares together reveals a fake image. With the property of unconditional security, VC is suitable for sending highly classified orders to a secret agent when computing devices may not be available.

The secret agent carried some shares, each with a pre-determined order, when departing to the hostile country. When the headquarter decides to execute a specific order, it can simply send another share to the agent so that the agent can recover what the order is. We can see that it would be terrible if the dispatched share cannot be verified due to a cheater's attack.

A VCS would be helpful if the shares are meaningful or identifiable to every participant. A VCS with this extended characteristic is called extended VCS (EVCS). EVCS is like a -VCS except that each share displays a meaningful image, which will be called *share image* hereafter. Different shares may have different share images. At first glance, it seems very difficult to cheat in EVCS because the cheater does not know the share images that appear on the genuine shares and, thus, has no information about the distributions of black and white pixels of the share images. This information is crucial for cheating in VC.

A VCS scheme is a 6-tuple (n, m, s, v, a, d):

It assumes that each pixel appears in *n* versions called shares, one for each transparency. Each share is a collection of *m* black and white sub-pixels. The resulting structure can be described by a *n x m* Boolean Matrix S=[Sij] where Sij=1 if the *Jth* sub pixel in the ith share is black. Therefore, the grey level of the combined share, obtained by stacking the transparencies, is proportional to the Hamming weight H(V) of the OR-ed m-vector V. This grey level is usually interpreted by the visual system as

black if H(V)>d and

as

white if H(V) < di-am

for some fixed threshold dm and relative difference a>0 , the difference between the minimum H(V) value of a black pixel and the maximum allowed H(V) value for a white pixel is called the contrast of a VCS scheme.

VCS scheme where a subset is qualified if and only if its cardinality is k are called (k, n)-threshold VCS consists of two collections of *n x m* Boolean matrices S0 and S1, each of size r. To construct a while pixel, we randomly chooses a matrices in S1. The chosen matrix will determine color of the m sub-pixels in each one of the n transparencies. Meanwhile, the solution is considered valid if the following three conditions are met:

1) For any matrix S in 0, the "or" operation on any k of the n rows satisfies H(V) <= d-am.
2) For any matrix S in 1, the "or" operation on any k of the n rows satisfies H(V)>= d-am.
3) For any subset {*i*1, *i2*, …… *i*q} of {1, 2, ……n } with q<k, the two collection of *q x m* matrices.

But obtaining by restricting each *n x m* matrix in St(where t=*f*0, 1*g)* two rows *i*1, *i*2, ….*i*q are indistinguishable in the sense that they contains exactly the same matrices with the same frequencies. In other words, any *q x n* matrices S0 -> B0 and S1 -> B1 are identical up to a column permutation. Condition (1) and (2) defines the contrast of VCS. Condition (3) states the security property of (k, n) – threshold VCS. Should we have not been given k shares of the secret image, we cannot gain any hint in deciding the color of out pixel, regardless of the amount of computation resource we have on hand.

Let us consider an instance of (3, 3) – threshold VCS construction where each pixel is divided into 4 sub-pixel (m=4). According to the definition, S0 and S1 are defined as the following:

S0= { all matrices obtained by permuting the columns of { 0 0 1 1},
{0 1 0 1}, {0 1 1 0} }

S1= { all matrices obtained by permuting the columns of {1 1 0 0},
{1 0 1 0}, {1 0 0 1} }

In order to encode a white pixel, the dealer needs to randomly choose one matrix from S0 to construct the sub-pixels in three shares accordingly. Meanwhile, to encode a black pixel, the dealer needs to randomly pick one matrix from S. It is not hard to verify that this construction will yield a

relative contrast of 0.25. That is, the encoding of a black pixel needs all 4 black sub-pixel, Where a white pixel needs 3 black sub-pixels and 1 white sub-pixel. Therefore, when the three shares stack together, the result is either dark grey, which we use to represent while , or completely black, which we use to represent black.

The Model

What is Visual Cryptography Scheme?

In 1994 by Noor and Shamir who introduced a simple but perfectly secure way that allows secret sharing without any Cryptographic computation, which they termed as Visual Cryptography Scheme (VCS). Their simplest Visual Cryptography Scheme is given by the following setup.

A secret image consists of a collection of black and white pixels where each pixel is treated independently. To encode the secret, we split the original image into n modified versions (referred as shares) such that each pixel in a share now subdivides into *m* black and white sub-pixels. To decode the image, we simply pick a subset S of those *n* shares and Xerox each of them onto a transparency. It is divided into 4 shares, which is denoted by Q containing at least one of the three sets {1, 2}, {2, 3} or {3, 4}.

Then the qualified sets are exactly the following:

E*qual* = { {1, 2}, {2, 3}, {3, 4}, {1, 2, 3}, {1, 2, 4}, {1, 3, 4}, {2, 3, 4}, {1, 2, 3, 4} }.

How are shares divided?

We have used halftone Visual Cryptography to achieve Visual Cryptography via half- toning. Based on the blue-noise dithering principles, the proposed method utilizes the void and cluster algorithm to encode a secret binary image into *n* halftone shares (images) carrying significant visual information.

How is Cheating done?

The issue of cheating by dishonest participants, called cheaters, in visual cryptography is that, cheating is possible when the cheaters form a coalition in order to deceive honest participants. At the outset two simple cheating prevention visual cryptographic schemes where proposed by G.B.Hond, T.G.Chen, and D.S.Tsai.

How is Decoding done?

Along with this basic setup, Noor and Shamir also proposed (k, n) threshold model as its extension. This extended scheme is constructed such that any k shares can be stacked together to reveal the original secret.

Modular Description

This system deals with security during transmission of images. In Cheating Prevention in Visual Cryptography the following are the main modules and each module is explained below.

- Security & Login Module
- Encoding the Image
- Decoding the Image
- Verification of Images
- User Interface & Manual

Security & Login Module:

This module deals with ensuring that only authorized users can access to this application. A database with user-id and password is used to validate the User entry. The input user-id is validated for a minimum of four characters. The application the opens into the application screen on validation.

Encoding the Image:

This system uses the Secret Sharing Scheme, which is a method for sharing a secret among a set p of p participants. The secret is encoded into n pieces called shares each of which is given to a distinct participant. Certain qualified subsets of participants can recover the secret by pooling together their information, whereas forbidden subsets of participants have information on the secret. The specification of the qualified sets and the forbidden sets is called access structure. A special kind of secret sharing scheme are Visual Cryptography Schemes (VCSs), that is, schemes where the secret to share is an image and the shares consist of Xeroxed transparencies which are stacked to recover the shared image. In this paper we analyze the relationship between secret sharing schemes and VCSs, focusing our attention on the amount of randomness required to generate the shares. We show how to transform a secret sharing scheme for a given accessness of the original scheme. An important consequence of this transformation is that lower bounds on the randomness of visual cryptography schemes apply to general secret sharing schemes. Our randomness preserving transformation has also been applied to derive a new upper bound on the randomness k, n threshold VCSs which dramatically improves on the previously known bounds. All VCSs obtained by applying our randomness preserving transformation allow a perfect reconstruction of black pixels.

Pixel		Share 1	Share 2	Result
	P = ½			
	P = ½			

Decoding the Image:

The secret image consists of a collection of black and white pixels. To construct shares of an image for participants, we need to prepare two collections, which consist of Boolean matrices. A row in a matrix corresponds to sub-pixels of a pixel, where 0 denotes the white sub-pixel and 1 denotes the black sub-pixel. For a white (or black) pixel in the image, we randomly choose a matrix from (or, respectively) and assign row of to the corresponding position of share. Each pixel of the original image will be encoded into pixels, each of which consists of sub-pixels on each share. Since a matrix in and constitutes only one pixel for each share. For security, the number of

matrices in and must be huge. For succinct description and easier realization of the VC construction, we do not construct directly. Instead, we construct two basis matrices and then let and be the set of all matrices obtained by permuting columns of and , respectively.

Two collections (multi-sets) of Boolean matrices constitute a -VCS if there exist a value and a set satisfying the following:

1) Any qualified set can recover the secret image by stacking their shares.

2) Any forbidden set has no information on the secret image.

Formally, the two collections, of matrices obtained by restricting each matrix in to rows, are indistinguishable in the sense that they contain the same matrices with the same frequencies.

In visual cryptography scheme, the carriers (share-image holders) will bring their shares to the target place. The decoder will then stack the share images to find the original image.

Verification of Images

In existing visual cryptography scheme, the carriers (share-image holders) will bring their shares to the target place; the decoder will then stack the share images to find the original image. Here, there is no guarantee that carriers would not have changed or faked their images. This is the problem statement as known.

In the proposed method, the decoder should be able to find whether the share images brought to the target place are faked or not. Also, each share-holder should be able to verify that other share-holder images are faked or not. In order to achieve this, the proposed method creates a verification image for each share holder.

Each shareholder should bring this verification image also to the target place. In the target place, before decoding, each share-holder will verify the other share-holders image. Then, the decoder will also check each share-holders image.

In the proposed method, during encoding, a verification image is specified. The
share-images are created in such a way that stacking one share and other share-holders verification-share will reveal the verification image. Suppose there are two share- holders. During encoding, two images for each share-holder are created. One of them is secrete share and other is verification share. So totally four images will be created during encoding. In the target place, the decoder will know what verification image should be displayed during stacking. The decoder will check all verification images by combining each secret share with every other verification share.

In our implementation, we have taken (2,2) viscrypt scheme. that is, there will be two share-holders and both two of them will participate in stacking, as u know.

To do automated verification, we have defined the options called flag1 and flag2 will be enabled after verification only. Also note that the message box should come automatically showing whether the verification is successful or not. This cannot be done automatically. The encoding and decoding can be done by computer. But the verification can be done by humans only, by manually seeing whether the verification image that came is correct or not. that is, verification can be done by visually only.

Hence it is called visual cryptography. We can't automate the verification part. That is the reason we have placed the Flag1 and Flag2 buttons. The receiver (the person who is decoding) should see the verification image, check it and press these two buttons. He would press the buttons only after confirming that correct verification images are received. After verification, decoding can be done usually.

Regarding cheating and fake shares: In the existing method, one of the participants would cheat initially, by converting his verification share into a fake share. During verification, the receiver can identify it whether that verification share is genuine or not. If the participant has cheated, then the correct verification share cannot be visually identified by the receiver. If it is not correctly identified, the receiver would stop verifying other shares and decoding. In our implementation, we have to run [MaliciousParticipant.java]. We have to select any one verification share and convert it into fake share. During verification, first show the demo with faked shares and then with original verification shares. an verification image is faked by randomly choosing a pixel for converting it into white pixel.

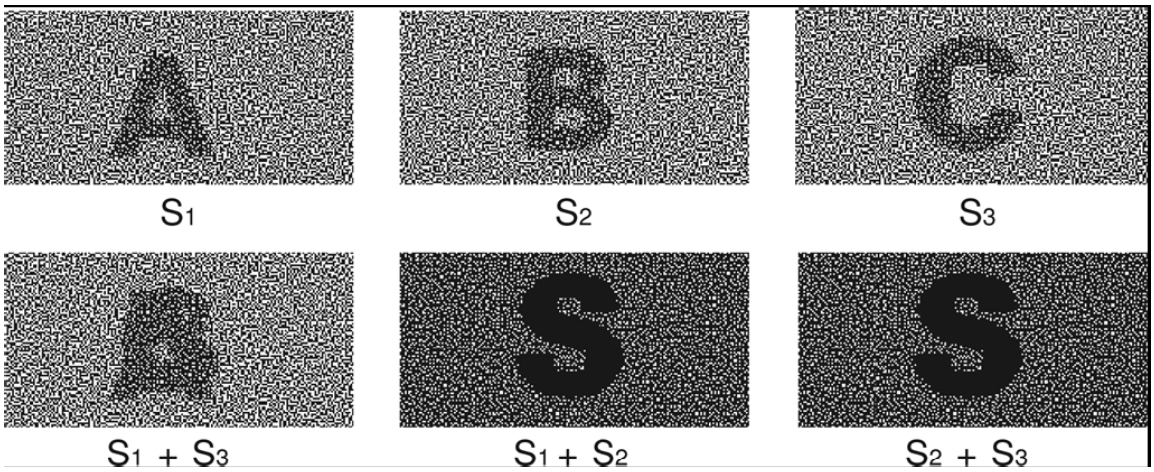
The Visual Cryptography Scheme can be illustrated as follows:

The secret image consists of a collection of black and white pixels. To construct shares of an image for participants, we need to prepare two collections, which consist of Boolean matrices. A row in a matrix corresponds to sub-pixels of a pixel, where 0 denotes the white sub-pixel and 1 denotes the black sub-pixel. For a white (or black) pixel in the image, we randomly choose a matrix from (or, respectively) and assign row of to the corresponding position of share.

Each pixel of the original image will be encoded into pixels, each of which consists of sub-pixels on each share. Since a matrix in and constitutes only one pixel for each share. For security, the number of matrices in and must be huge. For succinct description and easier realization of the VC construction, we do not construct and directly. Instead, we construct two basis matrices and then let and be the set of all matrices obtained by permuting columns of and, respectively. Two collections (multi-sets) and of Boolean matrices constitute a – VCS.

Each white pixel in the original image is split into two of the same small blocks that have half black and white pixels. When these two blocks are overlapped, they line up exactly, and the result is a light-colored block (with half black and half white pixels). Each black pixel in the original logo is split into two complementary small blocks. When these two blocks are overlapped, the result is a completely black box.

If each pixel in the original image is split randomly as described above, then each individual share is a totally random collection of blocks. Only when the shares are combined is any information revealed about the original image.



Extended VCS

UML diagrams for Cheating Prevention in Visual Cryptography

Class Diagram

A class diagram shows a set of classes, interfaces and collaborations and their relationships. These diagrams are the most common diagram found in modeling object-oriented systems.

Common Properties

A class diagram is just like as special kind of diagram and shares the same properties as all other diagrams. But it differs from all other diagrams in its contents.

Contents

Class diagram commonly contains the following things.

Class

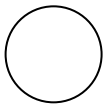
Class is a description of a set of objects that share the same attributes, operations, relationships and semantics. A class implements one or more interfaces. Graphically, a class is rendered as a rectangle, usually including its name, attributes and operations as shown in fig.

Window
Origin: char
open () close () move() display()

Interface

Interface is a collection of operations that specify a service of a class or component. An interface describes the

externally visible behavior of that element. An interface might represent the complete behavior of a class or component. Graphically, an interface is rendered as a circle together with its name as shown in fig.

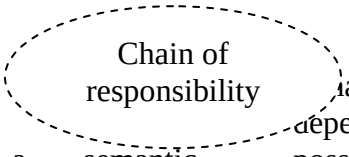


Interface

Collaboration

Collaboration defines an interaction and is a society of roles and other elements that work together to provide some cooperative behavior. So collaborations have structural as well as behavioral, dimensions. These collaborations represent the implementation of patterns that make up a system. Graphically, collaboration is rendered as an ellipse with dashed lines, usually including only its name as shown in fig

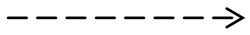
.



Collaboration

Dependency

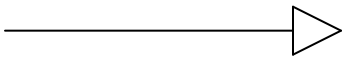
Dependency is a semantic relationship between two things in which a change to one thing may affect the semantics of the other thing. Graphically, a dependency is rendered as a dashed line, possibly directed, and occasionally including a label as shown in fig.



Dependency

Generalization

A generalization is a specialization / generalization relationship in which objects of the specialized element (child) are substitutable for objects of the generalized element (parent). In this way, the child shares the structure and the behavior of the parent. Graphically, a generalization relationship is rendered as a solid line with a hollow arrow head pointing to the parent as shown in fig.



Generalization

Association

An association is a structural relationship that describes a set of links, a link being a connection among objects. Aggregation is a special kind of association, representing a structural relationship between a whole and its parts. Graphically, an association is rendered as a solid line,

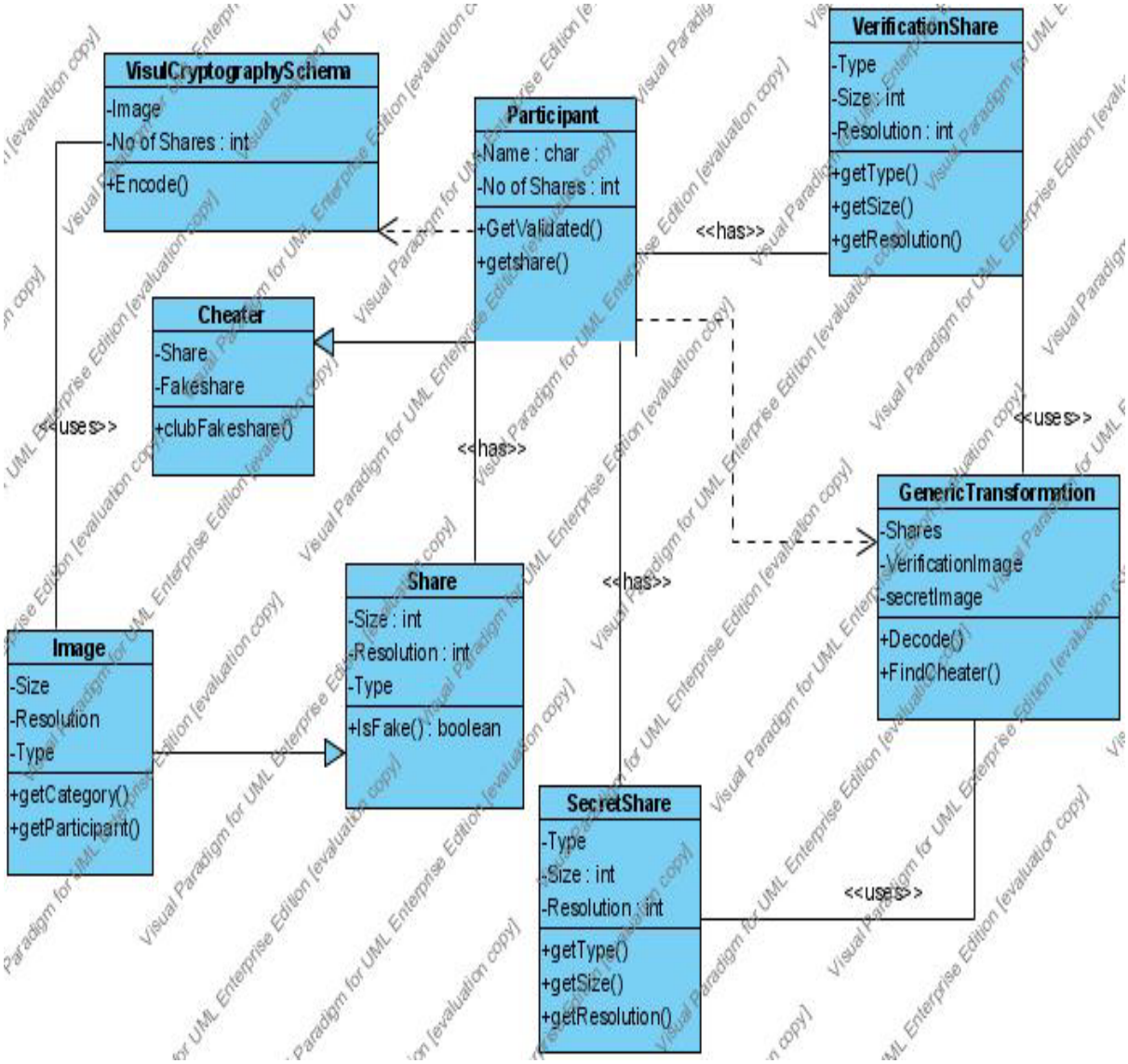
possibly directed, occasionally including a label as shown in fig.

0….1 *

Employer Association — Employee

Class diagram of Cheating Prevention in Visual Cryptography

Class diagram of Cheating Prevention in Visual Cryptography



It contains classes such as

- Visual Cryptography Schema
- Participant
- Verification Share
- Cheater
- Share
- Image
- Secret Share
- Generic Transformation

And the description is as follows

Visual Cryptography Schema class Description

Class	Visual Cryptography Schema
Description	Visual Cryptography Schema class is the one which is used inorder to divide the secret image and verification image into shares which each participant holds.
Attribute1 : Image Description : This specifies the image which it has to divide into shares. Attribute2 : Shares Description : This specifies the number of shares it is going to divide into. Operation : Encode() Description : Encodes (Divides) the image into shares.	

Participant class Description

Class	Participant
Description	Participant is the one who holds the secret share and Verification share which are divided by Visual cryptography Schema class
Attribute1 : Name Description : This specifies the name of the participant. Attribute2 : Shares Description : This specifies the number of shares a participant is holding	
Operation1 : GetValidated() Description : This operation specifies that every participant has to be validated. Operation2 : Getshare() Description : This specifies the shares a participant should receive.	

Verification Share Class Description

Class	Verification Share
Description	Verification Share is the one which is given to each and every participant and every verification share is different from other.
Attribute1 : Type Description : This specifies the type () of the image. Attribute2 : Size Description : This specifies the size of the image. Attribute3 : Resolution Description : This specifies the Resolution of the image. .	
Operation1 : GetType() Description : This operation reads the type of image. Operation2 : GetSize() Description : This operation reads the size of the image Operation3 : GetResolution Description : This operation reads the resolution of the image.	

Cheater Class Description

Class	Cheater
Description	Cheater is one of the participant who has a genuine share and verification share and who want to cheat the group.
Attribute1 : Share Description : This specifies the share of the cheater. Attribute2 : Fake Share Description : This specifies the Fake share developed by the cheater.	
Operation1 : ClubFakeShare() Description : This operation specifies that the cheater is going to club the fake share	

Share Class Descriptor

Class	Share
Description	Share is a part of the image which a participant holds.

Attribute1 : Type Description : This specifies the type (Format) of the share. Attribute2 : Size Description : This specifies the size of the share Attribute : Resolution. Description : This specifies the resolution of the share.
Operation1 : IsFake() Description : This specifies whether the share is a genuine one or not..

Image Class Description

Class	Image
Description	Image is the one which is divided into shares and distributed to each and every participant.
Attribute1 : Type Description : This specifies the type (Format) of the image. Attribute2 : Size Description : This specifies the size of the image. Attribute : Resolution. Description : This specifies the resolution of the image.	
Operation1 : GetCategory() Description : This specifies the category to which the image is belonging either secet image or verification image. Operation2 : GetParticipant() Description : This specifies to which participant the image is belonging to.	

Secret Share Class Description

Class	Secret Share
Description	Secret Share is the one which is given to each and every participant and every secret share is different from other.

Attribute1 : Type Description : This specifies the type () of the share. Attribute2 : Size Description : This specifies the size of the share. Attribute3 : Resolution Description : This specifies the Resolution of the share.
Operation1 : GetType() Description : This operation reads the type of share. Operation2 : GetSize() Description : This operation reads the size of the share. Operation3 : GetResolution Description : This operation reads the resolution of the share.

Generic Transformation Class Description

Class	Generic Transformation
Description	Generic Transformation is the one which do the decoding part of the secret image.
Attribute1 : Shares Description : This specifies the shares it has to which verification has to be performed. Attribute2 : Verification image Description : This specifies the verification image with which it performs the verification. Attribute3 : Secret image Description : This specifies the secret image with which it performs the verification.	
Operation1 : Decode() Description : This operation decodes the actual secret image. Operation2 : FindCheater() Description : This operation finds out who is the cheater.	

Use Case Diagram

A use case diagram shows a set of use cases and actors and their relationships. Use-case diagrams address the static use-case view of a system. These diagrams are especially important in organizing and modeling the behaviors of a system.

Common properties

A use-case diagram is a just a special kind of diagram and shares the same common properties as do all other diagrams. A use-case diagram differs from all other kinds of diagrams in its particular content.

Contents

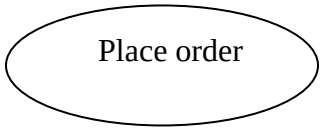
Use-case diagram commonly contain

Use-case

Use case is a description of set of sequence of actions that a system performs that yields an observable result of value to a particular actor. A use case is used to structure the behavioral things in a model. A use case is realized by collaboration. Graphically a use case is rendered as an ellipse with solid lines, usually including only its name, as shown below.

Actor

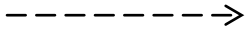
A coherent set of roles those users of use-cases play when interacting with the use cases.



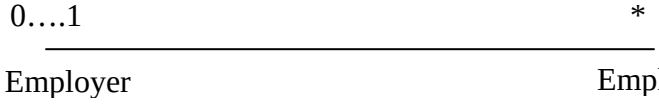
Use-case

Relationships

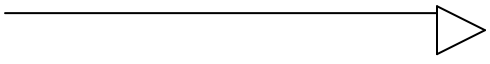
Dependency is a semantic relationship between two things in which a change to one thing may affect the semantics of the other thing. Graphically, a dependency is rendered as a dashed line, possibly directed, and occasionally including a label as shown in fig.



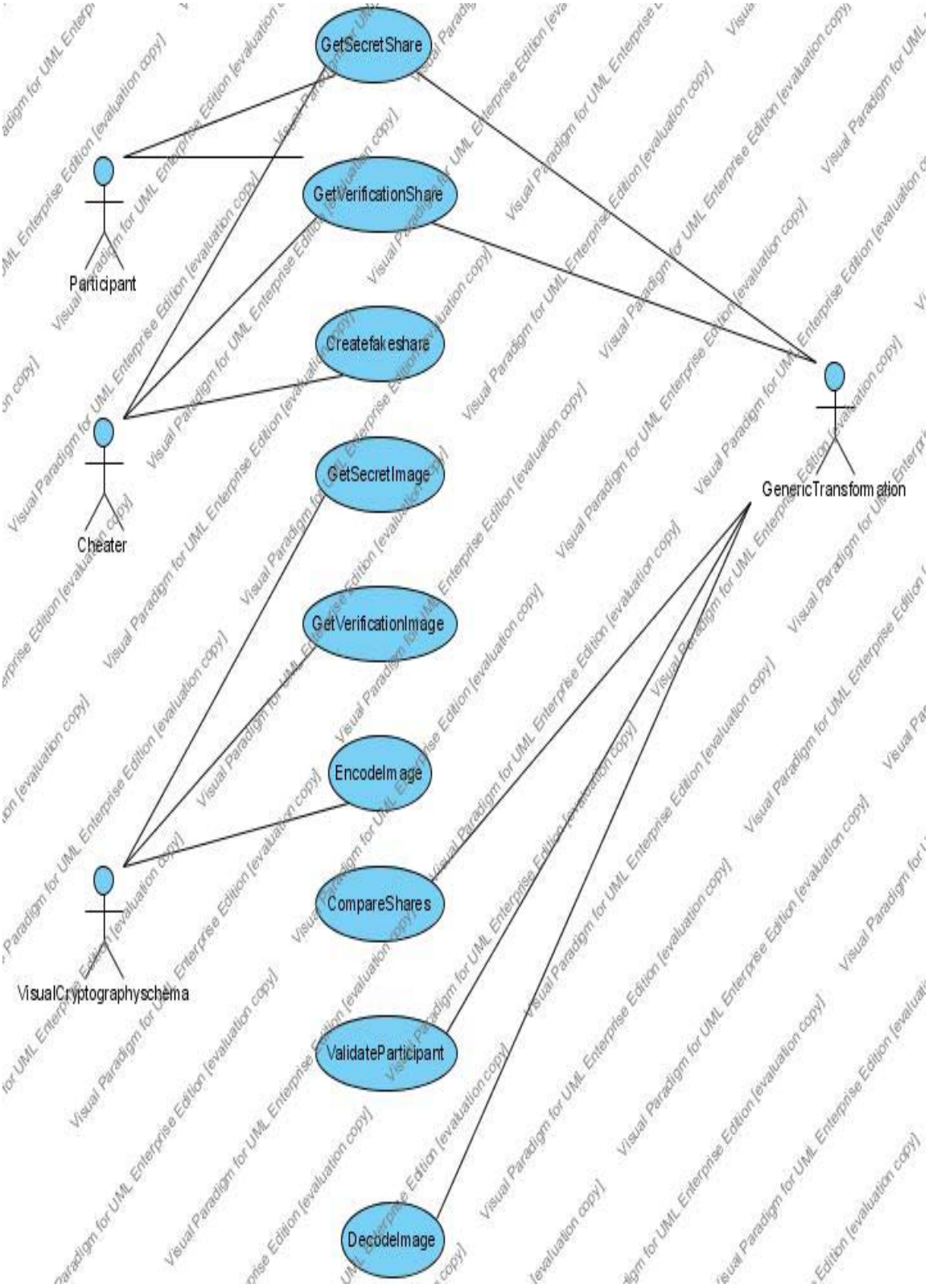
Association is a structural relationship that describes a set of links, a link being a connection among objects. Aggregation is a special kind of association, representing a structural relationship between a whole and its parts. Graphically, an association is rendered as a solid line, possibly directed, occasionally including a label as shown in fig.



Generalization is a specialization / generalization relationship in which objects of the specialized element (child) are substitutable for objects of the generalized element (parent). Graphically, a generalization relationship is rendered as a solid line with a hollow arrowhead pointing to the parent as shown in fig.



Like all other diagrams, use-case diagram may contain notes and constraints. Use-case diagram may also contain packages, which are used to group elements of your model into larger chunks.
Use Case diagram of Cheating Prevention in Visual Cryptography



Use Case diagram of Cheating Prevention in Visual Cryptography

It consists of use-cases

- Get Secret Share
- Get Verification share
- Create Fake Share
- Get Secret Image
- Get Verification Image
- Encode Image
- Compare Shares
- Validate Participants
- Decode Image.

Actors:

- Participant
- Cheater
- Visual Cryptography Schema
- Generic Transformation

Get Secret Share Use case Description

Use-case 1	: Get Secret Share
Actor	: Participant
Purpose	: Each and every participant holds a secret share.
Description	: Gets a secret share which has been divided from the secret image.

Get Verification Share Use case Description

Use-case 1	: Get Verification Share
Actor	: Participant
Purpose	: Each and every participant holds a verification share.
Description	: Gets a verification share which has been divided from the verification image.

Create Fake Share Use case Description

Use-case 1	: Create Fake Share
Actor	: Cheater
Purpose	: Cheater creates a Fake share for cheating others.

Description : Cheater having a genuine share creates a fake share for cheating.

Get Secret Image Use case Descriptor

Use-case 1 : Get Secret Image
Actor : Visual Cryptography Schema
Purpose : We take an secret image in order to divide it into shares.
Description : We will select an secret image and divides it into shares and allot each participant a secret share.

Get Verification Image Use case Description

Use-case 1 : Get Verification Image
Actor : Visual Cryptography Schema
Purpose : We take an verification image in order to divide it into verification shares .
Description : We will select an Verification image and divide it into verification shares and allot each participant a verification share.

Encode Image Use case Description

Use-case 1 : Encode Image
Actor : Visual Cryptography Schema
Purpose : This is used for encoding the secret image and verification image into shares.
Description : This is used for encoding the secret image and verification image into shares.

Compare Share Use case Description

Use-case 1 : Compare Shares
Actor : Generic Transformation
Purpose : Comparing of shares is for verification purpose
Description : We compare the shares of participants for the authorization purpose.

Validate Participant Use case Description

Use-case 1 : Validate Participant
Actor : Generic Transformation
Purpose : Validation is done to find out whether the participant is honest or not.
Description : We do the validation process to find out if the participant is a cheater or not.

Decode Image Use case Description

Use-case 1 : Decode Image

Actor	: Generic Transformation
Purpose	: This one is for Decoding the secret image.
Description	: If there is no cheating, decoding process will be successful.

Component Diagram

A component diagram shows the organizations and dependencies among a set of components. The component diagram emphasizes the static implementation view of a system.
The implementation view of a system encompasses the components and files that are used to assemble and release the physical system.

Common Properties

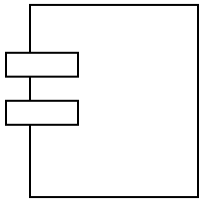
A component diagram is just a special kind of diagram and shares the same common properties as all the diagrams but they in terms of contents.

Contents

Component diagram commonly contains

Component

A component is a physical and replaceable part of a system that conforms to and provides the realization of a set of interfaces. Graphically, a component is rendered as a rectangle with tabs as shown in Fig.



Interface

An interface is a collection of operations that are used to specify a service of a class or component. Graphically it is rendered as a circle as shown in Fig.

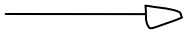
Dependency

A dependency is a using relationship that states that a change in specification of one thing may affect another thing that uses it, but not necessarily the reverse. Graphically, a dependency is rendered as a dashed line as shown in Fig.

Generalization ----------

A generalization is a relationship between a general thing i.e., (called the

super class or parent) and a more specific kind of that thing i.e., (called the subclass or child). Generalization is sometimes called an "is-a-kind-of" relationship. Graphically, generalization is rendered as a solid directed line with open arrowhead as shown in Fig.

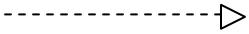


Association

An association is a structural relationship that specifies that objects of one thing are connected to objects of another. Graphically, an association is rendered as a solid line as shown in Fig.

Realization

A realization is a semantic relationship between classifiers in which one classifier specifies a contract that another classifier guarantees to carry out. Graphically, a realization is rendered as a dashed directed line with a large open arrowhead as shown in Fig.



Deployment diagram

A deployment diagram shows the configuration of run-time processing nodes and the components that live on them. Deployment diagrams address the static deployment view of architecture. They are related to component diagrams.

Common properties

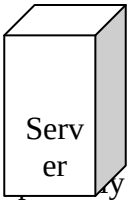
A deployment diagram is a just a special kind of diagram and shares the same common properties as do all other diagrams. A deployment diagram differs from all other kinds of diagrams in its particular content.

Contents

Deployment diagrams commonly contain

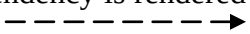
Node

A node is a physical element that exists at run time and represents a computational resource, generally having at least some memory and often, processing capability. Graphically, a node is rendered as a cube, usually including only its name, as in below figure.

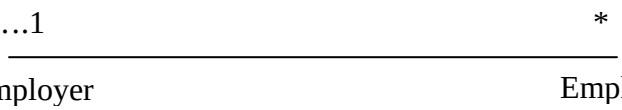


Relationships

Dependency is a semantic relationship between two things in which a change to one thing may affect the semantics of the other thing. Graphically, a dependency is rendered as a dashed line, possibly directed, and occasionally including a label as shown in fig.



An association is a structural relationship that describes a set of links, a link being a connection among objects. Aggregation is a special kind of association, representing a structural relationship between a whole and its parts. Graphically, an association is rendered as a solid line, possibly directed, occasionally including a label as shown in Fig.



Like all other diagrams, deployment diagrams may contain notes and constraints. Deployment diagrams may also contain components, each of which must live on some node. Deployment diagram may also contain packages, which are used to group elements of your model into larger chunks.

Sequence Diagram

The sequence diagram is an interaction diagram that emphasizes the time ordering of messages. Graphically, a sequence diagram is a table that shows objects arranged along the X axis and messages, ordered in increasing time, along the Y axis.

Common Properties

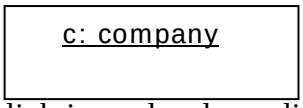
The sequence diagram is just like as special kind of diagram and shares the same properties as all other diagrams. But it differs from all other diagrams in its contents.

Contents

Sequence diagram commonly contains the following things

Objects

Objects are typically named or anonymous instances of class. But may also represent instances of other things such as components, collaboration and nodes. Graphically, object is rendered as a rectangle with underlining its name as shown in Fig.



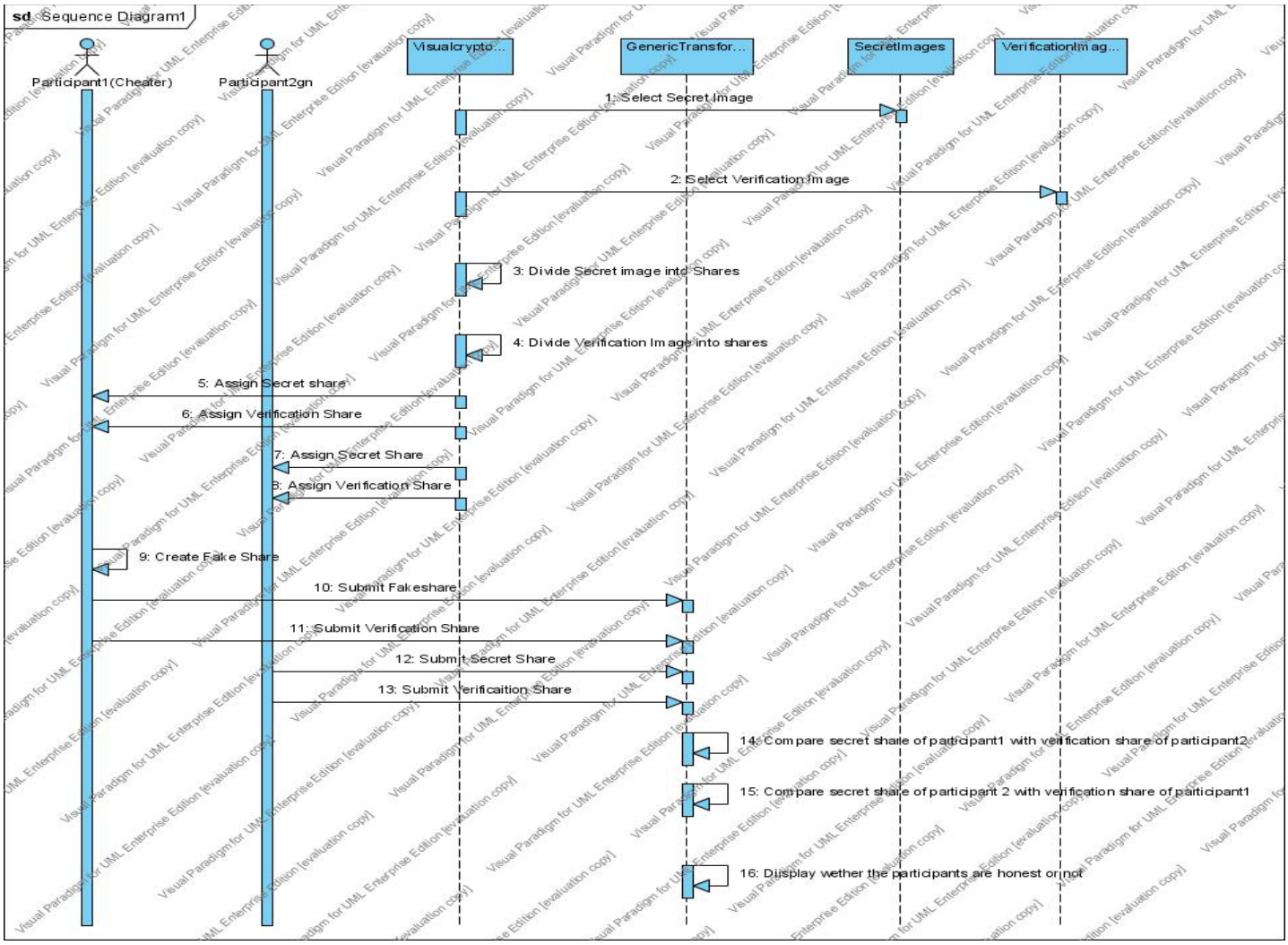
Links

A link is a semantic connection among objects i.e., an object of an association is called as a link. A link is rendered as a line as shown Fig.

Messages

A message is a specification of a communication between objects that conveys the information with the expectation that the activity will ensue.

Sequence Diagram of Cheating Prevention in Visual Cryptography



Sequence Diagram consists of objects

- Visual Cryptography Schema
- Generic Transformation
- Secret Images and Verification Image

Collaboration Diagram

The collaboration diagram is an interaction diagram that emphasizes the structural organization of objects that send and receive messages. Graphically, a collaboration diagram is a collection of vertices and arcs.

Common Properties

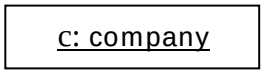
The collaboration diagram is just like as special kind of diagram and shares the same properties as all other diagrams. But it differs from all other diagrams in its contents.

Contents

Collaboration diagram commonly contains the following things

Objects

Objects are typically named or anonymous instances of class. But may also represent instances of other things such as components, collaboration and nodes. Graphically, object is rendered as a rectangle with underlining its name as shown in Fig.



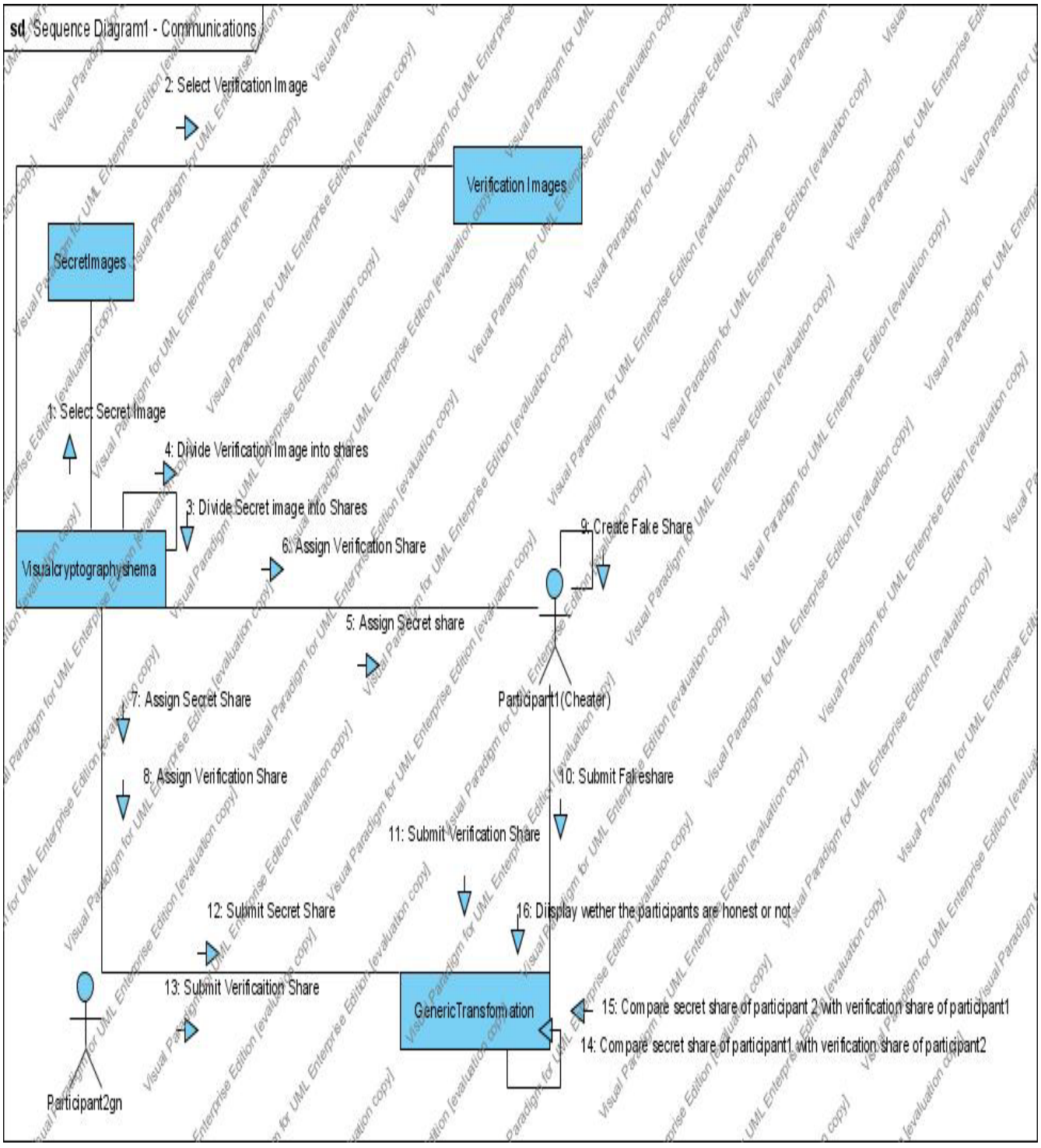
Links

A link is a semantic connection among objects i.e., an object of an association is called as a link. A link is rendered as a line as shown in Fig.

Messages

A message is a specification of a communication between objects that conveys the information with the expectation that the activity will ensue.

Collaboration Diagram of Cheating Prevention in Visual Cryptography:



Collaboration Diagram of Cheating Prevention in Visual Cryptography

Activity Diagram

An activity diagram shows the flow from activity to activity. The activity diagram emphasizes the dynamic view of a system.

Common Properties

An activity diagram is just a special kind of diagram and shares the same common properties as do all other diagrams but they differ in their content.

Contents

Activity diagram commonly contain

Activity states and Action states

An activity states is a kind of states in activity diagram; it shows an ongoing non-atomic execution within a state machine. An activity states can be further decomposed. An action states are States of the system, each representing the execution of an Action. An action states can't be further decomposed.

The activity and action states are rendered as

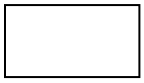
Transition

A transition specifies the path from one action or activity state to the next action or activity state.

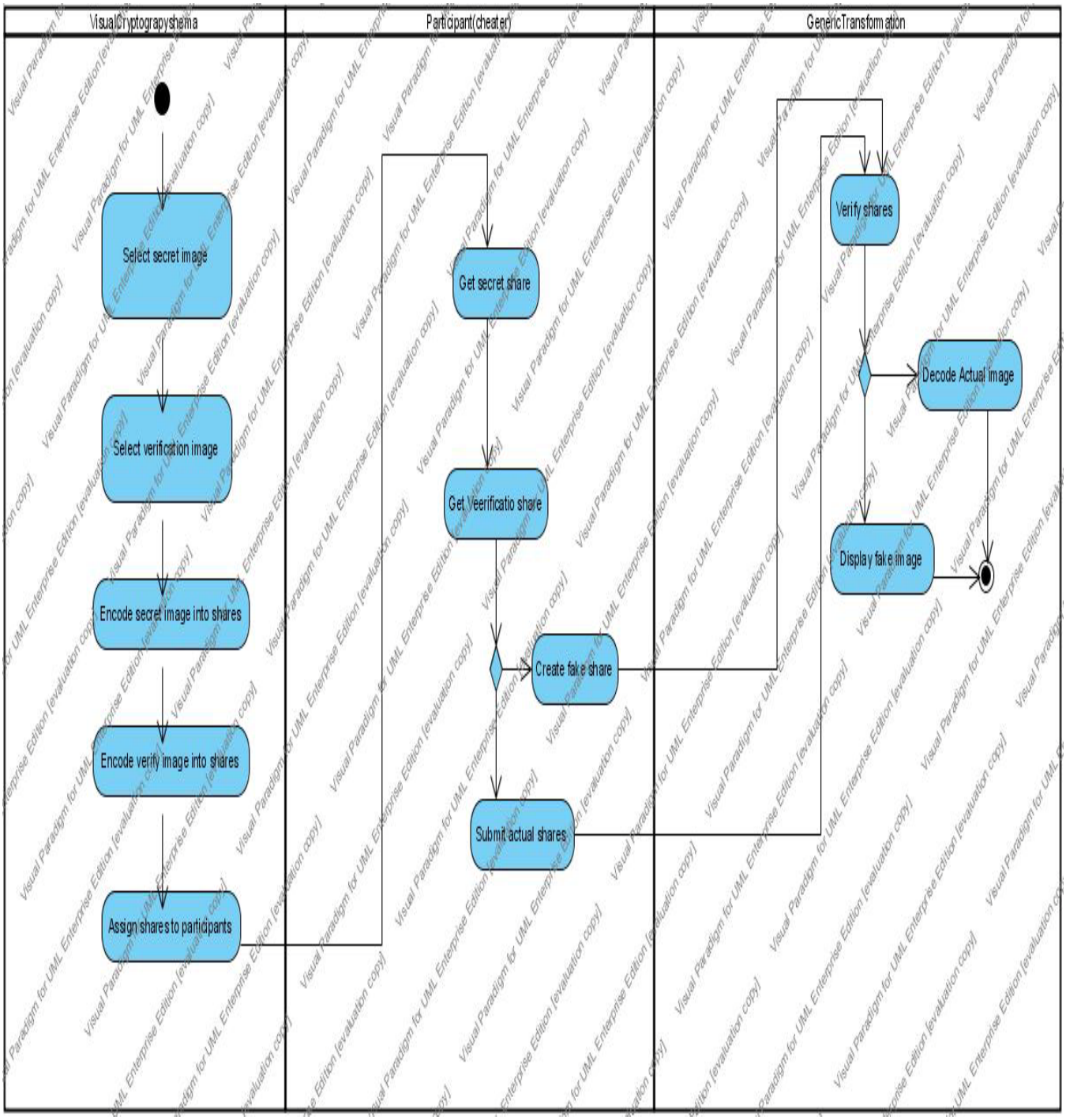
The transition is rendered as a simple directed line.

Object

An object is a concrete manifestation of an abstraction; an entity with a well defined boundary and identity that encapsulates state and behavior; an instance of a class. Objects may be involved in the flow of control associated with an activity diagram. The object is rendered as a rectangle.



Activity Diagram of Cheating Prevention in Visual Cryptography



Activity Diagram of Cheating Prevention in Visual Cryptography

State chart Diagram

A state-chart diagram shows a state machine, consisting of states, transitions, events, and activities. The state-chart

diagram emphasizes the dynamic view of a system.

Common properties

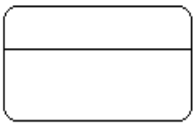
A state-chart diagram is just a special kind of diagram and shares the same common properties as do all other diagrams but they differ in terms of contents.

Contents

State-chart diagram commonly contains

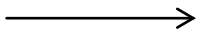
State

A state is a condition or situation in the life of an object during which it satisfies some condition, performs some activity, or waits for some event. Graphically, a state is rendered as a rectangle with rounded corners.



Transition

A transition is a relationship between two states indicating that an object in the first state will perform certain actions and enter the second state when a specified event occurs and specified conditions are satisfied. Graphically, a transition is rendered as a solid directed line.



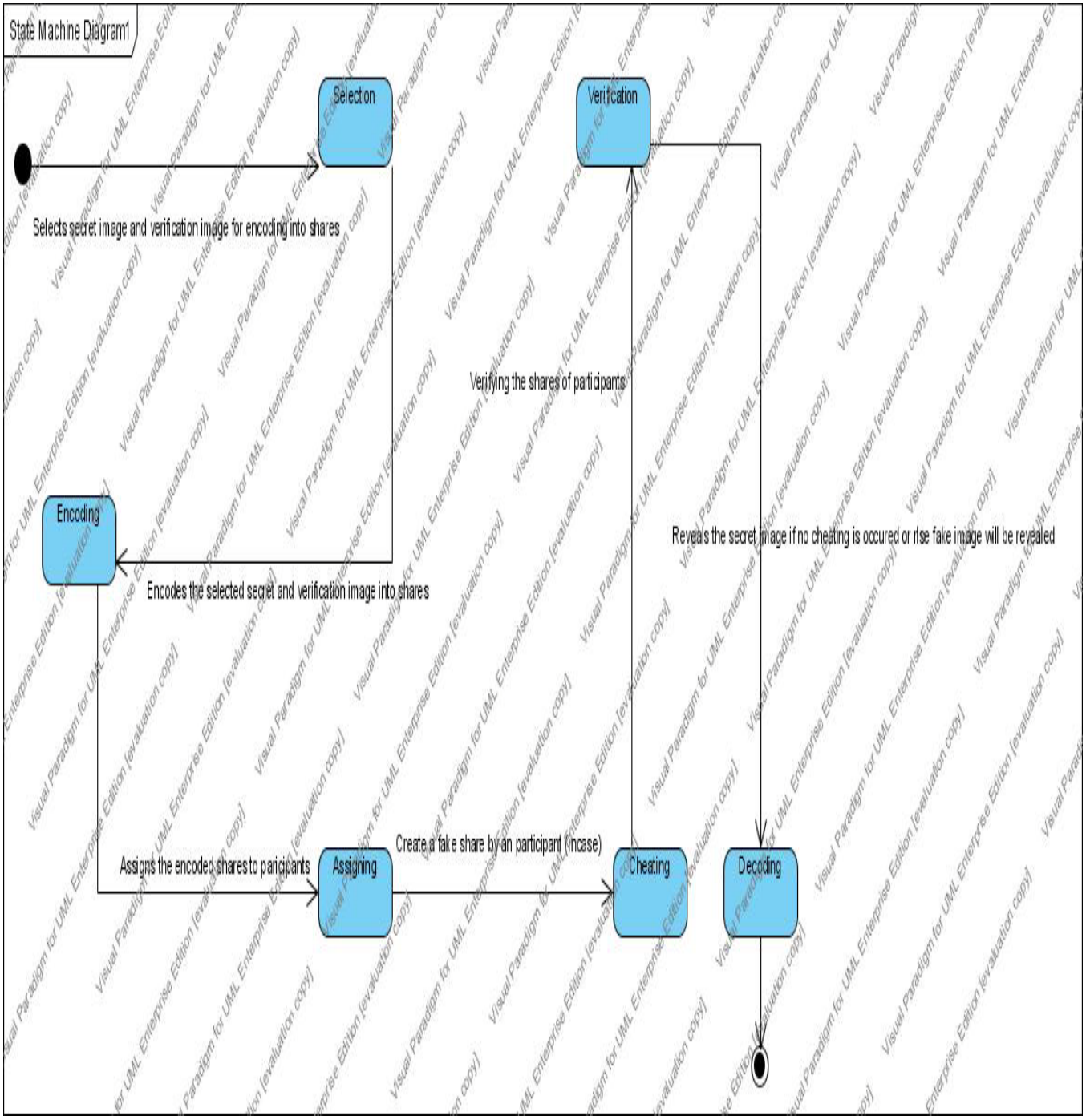
Event

An event is the specification of a significant occurrence that has a location in time and space.

Action

An action an executable atomic computation that results in a change in state of the model or the return of a value.

State Chart Diagram of Cheating Prevention in Visual Cryptography



State Chart Diagram of Cheating Prevention in Visual Cryptography

Testing:

Following are the some of the testing methods applied to this effective project:

Specification Testing

We can set with, what program should do and how it should perform under various conditions. This testing is a comparative study of evolution of system performance and system requirements.

Module Level Testing

In this the error will be found at each individual module, it encourages the programmer to find and rectify the errors without affecting the other modules.

Unit Testing

Unit testing focuses on verifying the effort on the smallest unit of software-module. The local data structure is examined to ensure that the date stored temporarily maintains its integrity during all steps in the algorithm's execution. Boundary conditions are tested to ensure that the module operates properly at boundaries established to limit or restrict processing.

Integration Testing

Data can be tested across an interface. One module can have an inadvertent, adverse effect on the other. Integration testing is a systematic technique for constructing a program structure while conducting tests to uncover errors associated with interring.

Validation Testing

It begins after the integration testing is successfully assembled. Validation succeeds when the software functions in a manner that can be reasonably accepted by the client. In this the majority of the validation is done during the data entry operation where there is a maximum possibility of entering wrong data. Other validation will be performed in all process where correct details and data should be entered to get the required results.

Recovery Testing

Recovery Testing is a system that forces the software to fail in variety of ways and verifies that the recovery is properly performed. If recovery is automatic, re-initialization, and data recovery are each evaluated for correctness.

Security Testing

Security testing attempts to verify that protection mechanism built into system will in fact protect it from improper penetration. The tester may attempt to acquire password through external clerical means, may attack the system with custom software design to break down any defenses to others, and may purposely cause errors.

Performance Testing

Performance Testing is used to test runtime performance of software within the context of an integrated system. Performance test are often coupled with stress testing and require both software instrumentation.

System Testing

Testing the entire system as a whole and checking for its correctness is system testing. The system is listed for dispensaries between the system and its original objectives. This project was effective and efficient.

Output Testing

After performing the validation testing, the next step is output testing of the proposed system since no system would be termed as useful until it does produce the required output in the specified format. Output format is considered in two ways, the screen format and the printer format.

User Acceptance Testing

User Acceptance Testing is the key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with prospective system users at the time of developing and making changes whenever required.

The flowing are the testing points:

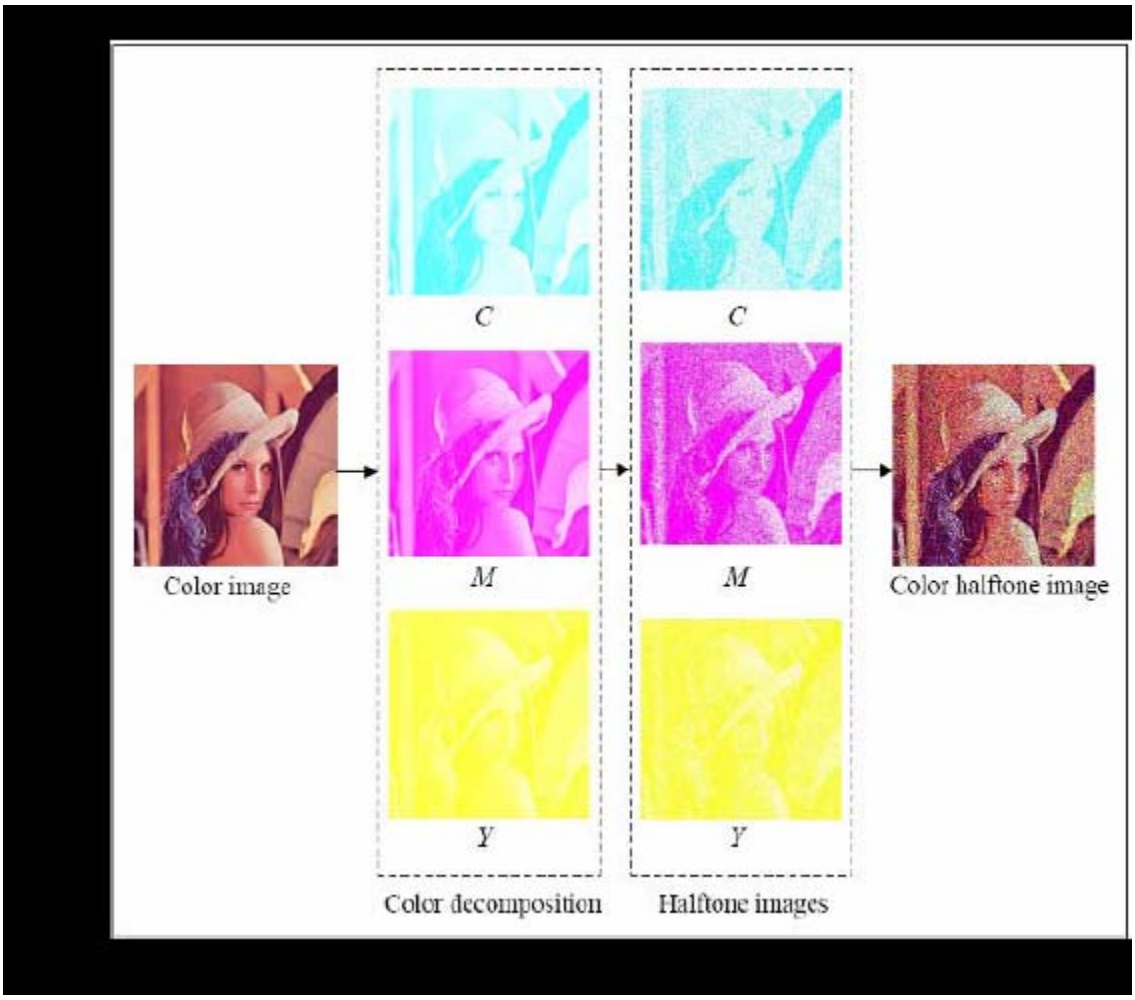
- Input Screen design
- Output Screen design
- Menu-driven System

Implementation is the process of bringing the developed system into operational use and turning it over to the user. The implementation of computer based system requires that test be prepared and that the system and its elements be tested in planned and structured manner.

All the above schemes can be used only to share the black and white secret images, but it is demand of time that schemes should also support color images. To meet this demand researches have been made to share the color images.

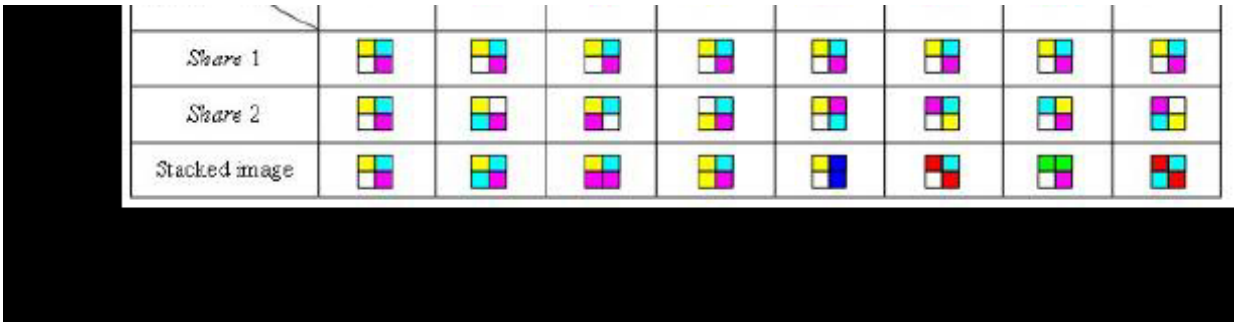
Color Visual Cryptography Scheme

Hou proposed three color VC methods where the same technique is used to decompose the color secret image into three separate images that are respectively colored cyan (C), magenta (M) and yellow (Y). Then the halftone technique is used to translate the three color images into halftone images. Finally, by combining the three halftone images, a color halftone image can be generated. The color halftone image generation process is shown in Fig.



Color decomposition

The color halftone image takes eight different colors to display: cyan, magenta, yellow, black, red, green, blue and white. The three methods proposed take the color halftone image as the secret image. Here, we focus on the second method and describe the details of this method. For each pixel of the color halftone image, the following process must be done. First, 2×2 blocks are built according to Share 1, and the four pixels C, M, Y and W are randomly permuted. Then, the number of blocks is calculated for Share 2 according to the color ratio of the four pixels with the coding table (Table below) referred to.



For example, if one pixel of the color halftone image is green, then the pixel's color ratio would be 100%, 0% and 100% for *C*, *M* and *Y*, respectively. Thus, block in *Share* 1 is the permutation of pixels: cyan, magenta, yellow and white. Then, the above information is applied, and the coding table will be referred to produce block of *Share* 2, where the permutation of the pixels is yellow, magenta, cyan and white. When all the pixels are done processed, two shares are produced. Each block of the two shares will be composed of *C*, *M*, *Y* and *W*. The secret image can be readily recognized visually when the two shares are stacked together.

The Proposed Scheme

There are four main procedures in the proposed scheme. The first procedure is color halftone transformation, where the color image is transformed to a color halftone image. The second procedure, pixel extraction process, extracts pixels from the color halftone image. Then, the following are encoding and decoding procedures, respectively. To generate the shares, two $N \times N$ cover images, named *CA* and *CB*, are used to encode the $N \times N$ secret image *SI* and make two $2N \times 2N$ shares called *Share* 1 and *Share* 2 . *Share* 1 will be a meaningful share that appears just like *CA*, and *Share* 2 will be also a meaningful share that looks just like *CB*. Finally, during the decoding procedure, the secret image can be easily reconstructed by stacking *Share* 1 and *Share* 2 together. There are two coding tables referred to in the encoding procedure: cover coding table (CCT) and secret coding table (SCT). As the names suggest, CCT is responsible for the encoding of the cover image, and SCT, on the other hand, is used to encode the secret image. The way SCT works in our new scheme is the same as it does in the second scheme of [5] (as shown in Table above).

Color Halftone Transformation and Pixel Extraction

Before encoding happens, this scheme applies color halftone transformation to produce color halftone images out of *CA*, *CB* and *SI*. Thus, *CA*, *CB* and *SI* are transformed into color halftone images *CA*', *CB*' and *SI*', respectively. The translation procedure is shown. Next, the pixel extraction procedure is utilized for reducing the size of the color halftone image. The
proposed scheme extracts some pixels from the color halftone image as important information for later coding. For each halftone image generated, the pixels from the odd-numbered rows, or
those from the even-numbered rows, can be extracted out to make the extracted image, which means the size of the extracted image is N*N/2. In such a way, *CA*', *CB*' and *SI*' are pixels extracted to generate *EA*, *EB* and *ES*. In other words, our new scheme can have the secret image restored with only half

of the pixels at hand. This helps both save storage space in the main memory and shorten the encoding time.

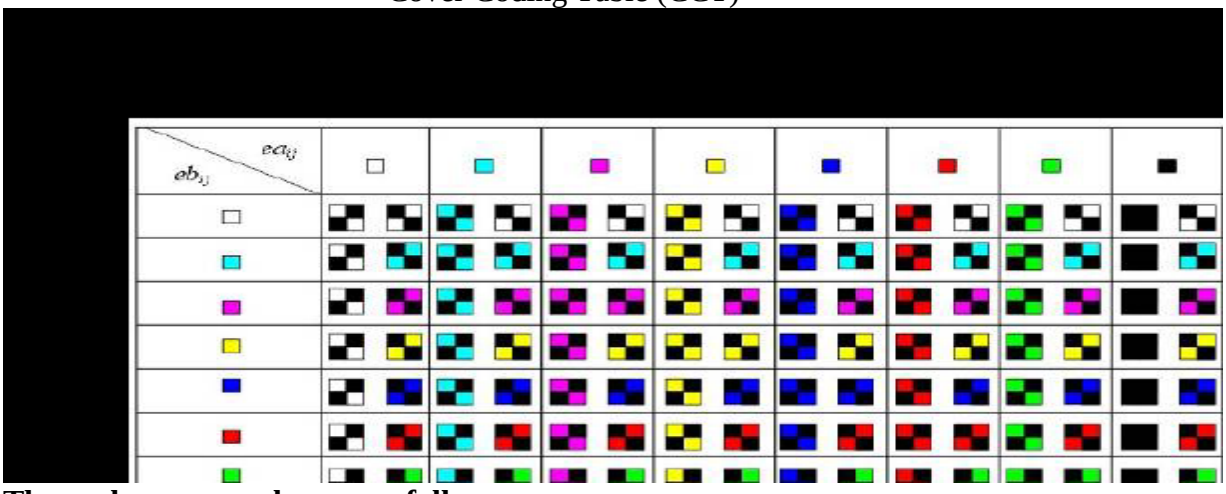
Encoding and Decoding

During the encoding procedure, our new scheme takes in two coding tables, cover coding table (CCT) and the secret coding table (SCT), respectively. CCT is to help with the encoding of the extracted cover image, and SCT is to help process the extracted secret image. SCT in our new scheme works the same way as Table. In the encoding procedure, the proposed scheme uses CCT to encode *EA* and *EB*, while *ES* is encoded by the SCT. In CCT, as shown in Table , the first row represents various color pixels in *EA* and the first column stands for various color pixels in *EA*. The intersections of the rows and the columns are the output blocks with the left side of the block belonging to *Share* 1 and the right side of the block belonging to *Share* 2. SCT, as shown in Table, has the same definition as it does. Each pixel from the extracted image is expanded to one 2×2 block. The expanded block is placed in one of the 2×4 block patterns. By
this way, the extracted image can produce a 2*N*×2*N* share.

After the color halftone transformation and pixel extraction, the proposed scheme has generated three color halftone images and extracted all the pixels it needs for *CA*, *CB* and *SI*, which are *EA*, *EB* and *ES*. As seen in Table, CCT is used to generate a 2×2 block from *EA*, and this block belongs to *Share* 1. As seen in Table as well, CCT is used to generate a 2×2 block from *EB*, and this block belongs to *Share* 2. Then, the color ratio of the pixel is analyzed according to its position in the extracted image.

According to the color ratio with Table referred to, two 2*2 blocks, namely block 1 and block 2, can be produced. The pattern is divided into two regions, region I and region II. Each region covers a 2*2 blocks area. Based on the position of pixel *eaij* or *ebij*, region I or
region II is replaced with a suitable block.

Cover Coding Table (CCT)



The replacement rules are as follows

(1). When pixels *eaij* and *ebij* are on an odd row (*I* mod 2 = 0), replace region I in pattern 1 with

block 1, and replace region I in pattern 2 with block 2. In contrast, if pixel *eaij* and *ebij* are on an even row (*i* mod 2 =1), replace region II in pattern 1 with block 1, and replace region II in pattern 2 with block 2. Now the encoding procedure for *EA* and *EA* is completed.

(2). For the encoding of *ES*, the proposed scheme needs to analyze the color ratio of the pixels. Then, according to the color ratio with the SCT (table) referred to, block 3 and block 4 can be generated. The position *esij* in *ES* is defined, where 0 <=*i*<*N* and 0<*j*<=*N/2*. When pixel *esij* is on an odd row (i.e. *i* mod 2 =0), replace region II in pattern 1 with block 3, and replace region II in pattern 2 with block 4. In contrast, if pixel *esij* is on an even row (i.e. *i* mod 2 =1), replace
region I in pattern 1 with block 3, and replace region I in pattern 2 with block 4. After completing pattern 1 and pattern 2, we put them in the matching positions in Share 1 and Share 2, respectively. When all the pixels of *EA*, *EB* and *ES* are done processed the production of *Share* 1 and *Share* 2 is completed. In the decryption process, we stack *Share* 1 and *Share* 2 together to reconstruct the secret image. Also, blocks representing *eaij* and *ebij* become black after the stacking, but will not affect the block which represents *esij*. Meanwhile, this can improve the
contrast of the secret image and make the image clearer.

Applications

Visual Cryptography schemes can decode concealed images based purely on human visual systems, without any aid from Cryptographic computation. This nice property gives birth to a wide range of encryption applications. In this section, we will discuss how VCS is used in applications such as E-Voting System, Financial documents and Copyright Protections.

1) Electronic-Balloting System

Now a days, most of the Voting is managed with Computer Systems. These Voting machines expected voters to trust them, without giving proof that they recorded each vote correctly. One way to solve this problem is to issue receipts to Voters to ensure them their votes are counted.

2) Encrypting Financial documents

The VCS principle can also be applied in transmitting confidential Financial documents over Internet. VCRYPT is an example of this type of system being proposed by Hawkes et al. VCRYPT can encode the original drawing document with a specified (k, n) VCS, then send each of the encoded n shares separately through Emails or FTP to the recipient.

Limitations

1) During transmission contrast digression and pixel expansion will taken place.
2) During transmission of secrete image the quality of the image will be disturbed.

Future Scope

Visual cryptography technique is used to make the data secure. Here the original data is divided into a number of shares which are sent through different communication channels from sender to receiver. Therefore the intruder has less chance to get the whole information. But still it is not so secured. This can be made more secure by introducing a symmetric key for both encryption and decryption process.

Using the key, the image is first encrypted then divided into a number of shares. If the intruder gets k number of shares s/he cannot be able to decrypt it if the key is not known to his/her. For key, a combination of character or number can be used. The change of higher bits make the image more blur, so the key can be applied on the higher bits of each pixels. A small image can also be used as a key. Let an image with size w1 X h1 is taken as a key where w1<w and h1<h. The original image is divided into blocks of w1 X h1. For each block, (w1, h1) th pixel is encoded with (w1, h1) th pixel of the key image. The reverse process will be applied for decryption.

Conclusion

The existing system the dealer or sender takes a secret image and encodes into shares. After encoding this shares are sent to participants. The receiver collects the shares and stack to get decoded secret image. Here no verification is done so easy cheating is done.

In this paper we proposed a system such that the dealer or sender takes one secret image and verification image. These two images are encoded into shares, after encoding sends one secret share and one verification share to the participants. Each participant verifies the share and other participant secret share reveals the secret image. In this way **cheating is avoided**.

In this paper we have proposed a technique of well known secret sharing on both black and white and color images. At the time of dividing an image into n number of shares we have used random number generator, which is a new technique not available till date. This technique needs very less mathematical calculation compare with other existing techniques of visual cryptography on color images. This technique only checks ‘1’ at the bit position and divide
that ‘1’ into (n-k+1) shares using random numbers. In most of our experimental results, each share reflects very little or even no information regarding the original image to human eye.
But the main drawback of the algorithm is in its number of loops. For n=6, k=5 and a 32 bit pixel with 50% ‘1’, number of loop operation required is 32. For n=6, k=4 with other conditions same, number of loop operation required is 48. For n=6, k=3 with other conditions same, number of loop operation required is 64.