IMPLEMENTING MODIFIED 3-BY-4 VCS SCHEME FOR GRAYSCALE IMAGES USING 4 SUBPIXEL LAYOUTS

Abhishek Garg[1]and Kishan Pal Singh[2]

[1]Department of Computer Engineering & Applications, Mangalayatan University, Aligarh, UP
[2]Department of Mechanical Engineering, Mangalayatan University, Aligarh, UPCorrespondingAuthorsEmail:[1]abhishek47722@gmail.com,[2]kishan.singh@mangalayatan.edu.in

ABSTRACT:

XNOR-based Modified 3-by-4 VCS scheme for secret sharing of images to a group of participants has been implemented with comparison of the results with earlier schemes. C Programming language based implementation requires to be done as an appreciation of the results as compared to earlier schemes. Same level of security as for other similar schemes for colored images has been developed for grayscale images. A 4 sub-pixel layout has been used for the implementation. Implementation of the concept has been done with less pixel expansion and distortion which is there in 4 sub-pixel layout based 3-by-4 VCS scheme.

KEYWORDS:

C language, Modified gray scale based 3-by-4 Visual Cryptography.

DOI Number: 10.14704/NQ.2022.20.15.NQ88586 **NeuroQuantology2022;20(15): 5819-5829**

INTRODUCTION:

Visual cryptography is a cryptographic technique which allows visual information (pictures, text, etc.) to be encrypted in such a way that decryption becomes a mechanical operation that does not require a computer.Plaintext is as an image.Encryption in Visual cryptography means creating image shares .Give the shares to the respective holders.Decryption – involving bringing together the appropriate combination and the human visual system.[1]

Thomas Monoth, BabuAnto P (2010) developed a scheme for transmission of fingerprints using secure channel using visual cryptography. A 2x2 VCS scheme algorithm has been developed to transmit fingerprints images through multiple shares such that k out of n participants is needed to obtain the original image (kxnvcs scheme). Any number less than k will not result in obtaining the secret image through shares. [2]

3-by-4 VCS scheme is used for Bio Metrics. Biometrics is the measurement and statistical analysis of people's unique physical and behavioral characteristics. [3]

Mask	Revealed color	Share1	Share2	Share3	Stacked image	Revealed color quantity
	(0, 0, 0)					(1/2, 1/2, 1/2)
	(1, 0, 0)					(1, 1/2, 1/2)
	(0, 1, 0)					(1/2, 1, 1/2)
	(0, 0, 1)					(1/2, 1/2, 1)
	(1, 1, 0)					(1, 1, 1/2)
	(0, 1, 1)					(1/2, 1, 1)
	(1, 0, 1)					(1, 1/2, 1)
	(1, 1, 1)					(1, 1, 1)

Fig1. 3-by-4VCS with 4 sub-pixel layout for grayscale images

eISSN 1303-5150

RELATED WORK:

Draw an image in some area,let us say write text using outtextxy function. Store image coordinate area through getimage function.

1. Generate share1 of sx4s size.

a)If image through getimage at (x,y) has White pixel;

store at(x',y'),(x',y'+1),(x',y'+2) & (x',y'+3) 0,1,1,1 respectively as per scheme 1; or 1,0,1,1 resp. as per scheme 2;or 1,1,1,0 resp. as per scheme 3 or 1,1,0,1 resp. as per scheme 4.

b) If image through getimage at (x,y) has Black pixel;

store at(x',y'),(x',y'+1),(x',y'+2) & (x',y'+3) 0,1,1,0 respectively as per scheme 1; or 0,0,1,1 resp. as per scheme 2;or 1,0,0,1 resp. as per scheme 3 or 1,1,0,0 resp. as per scheme 4.

2. Generate share 2 of sx4s size.

a)If image through getimage at (x,y) has White pixel;

store at(x',y'),(x',y'+1),(x',y'+2) & (x',y'+3) 0,1,1,1 respectively as per scheme 1; or 1,0,1,1 resp. as per scheme 2;or 1,1,1,0 resp. as per scheme 3 or 1,1,0,1 resp. as per scheme 4.

b) If image through getimage at (x,y) has Black pixel;

store at(x',y'),(x',y'+1),(x',y'+2) & (x',y'+3) 1,0,1,0 respectively as per scheme 1; or 0,1,0,1 resp. as per scheme 2;or 1,0,1,0 resp. as per scheme 3 or 0,1,0,1 resp. as per scheme 4.

3. Generate share 3 of sx4s size.

a)If image through getimage at (x,y) has White pixel;

store at(x',y'),(x',y'+1),(x',y'+2) & (x',y'+3) 0,1,1,1 respectively as per scheme 1; or 1,0,1,1 resp. as per scheme 2;or 1,1,1,0 resp. as per scheme 3 or 1,1,0,1 resp. as per scheme 4.

b) If image through getimage at (x,y) has Black pixel;

store at(x',y'),(x',y'+1),(x',y'+2) & (x',y'+3) 1,1,0,0 respectively as per scheme 1; or 0,1,1,0 resp. as per scheme 2;or 0,0,1,1 resp. as per scheme 3 or 1,0,0,1 resp. as per scheme 4.

Superimpose the three shares above by accessing first of the pixels at (x',y'),(x'',y'')&(x''',y''') through getpixel function and applying XNOR between them & subsequently printing the generated pixel at specified coordinates through putpixel; it generates the original image.A single share can't produce the image.[4]

PROPOSED WORK:

The implementation of the 3-by-4 VCS algorithm with output has been shown.

SOURCE CODE:

```
#include<stdio.h>
#include<graphics.h>
#include<conio.h>
#include<string.h>
#include<stdlib.h>
#include<math.h>
void main()
{
intgdriver=DETECT,gmode,x,y;
int x1,y1,x2,y2,x3,y3,x4,y4,scheme,val1,val2,val3=0,val,temp,count=0,save,i,flag=3;
char far *buf,*ptr;
unsigned size;
int color,color1,color2,color3;
char *str;
gets(str);
initgraph(&gdriver,&gmode,"c:\\turboc3\\bgi");
outtextxy(0,10,str);
```


eISSN 1303-5150

```
/* Taking image in buf with first two words of buf storing width and height of
area of screen holding the image*/
size=imagesize(0,10,300,30);
buf=(char far *)malloc(size);
//ptr=buf+2;
getimage(0,10,250,30,buf);
// getimage(0,5,100,10,buf);
//  putimage(0,15,buf,COPY_PUT);
//putimage(0,30,buf,COPY_PUT);
/* generate share1 */
x1=0,y1=40;
scheme=random(4);
for(x=0;x<=250;x++)
{
for(y=10;y<=30;y++)
//for(x=0,y=10;x<=350 && y<=20;x++,y++)
{
if(getpixel(x,y)==WHITE)
{     switch(scheme)
                    {case 0:
                            putpixel(x1,y1++,BLACK);
                            putpixel(x1,y1++,WHITE);
                            putpixel(x1,y1++,WHITE);
                            putpixel(x1,y1++,WHITE);
                     //     y1=y1+2;
                            break;
                    case 1:
                            putpixel(x1,y1++,WHITE);
                            putpixel(x1,y1++,BLACK);
                            putpixel(x1,y1++,WHITE);
                            putpixel(x1,y1++,WHITE);
                            break;
                    case 2:
                            putpixel(x1,y1++,WHITE);
                            putpixel(x1,y1++,WHITE);
                            putpixel(x1,y1++,WHITE);
                            putpixel(x1,y1++,BLACK);
                            break;
                    case 3:
                            putpixel(x1,y1++,WHITE);
                            putpixel(x1,y1++,WHITE);
                            putpixel(x1,y1++,BLACK);
                            putpixel(x1,y1++,WHITE);
                     }
}
else
{     switch(scheme)
                     {case 0:
                    putpixel(x1,y1++,BLACK);
```


eISSN 1303-5150

```
                putpixel(x1,y1++,WHITE);
                putpixel(x1,y1++,WHITE);
                putpixel(x1,y1++,BLACK);
                  //  y1=y1+2;
                        break;
                case 1:
                putpixel(x1,y1++,BLACK);
                putpixel(x1,y1++,BLACK);
                putpixel(x1,y1++,WHITE);
                putpixel(x1,y1++,WHITE);
                break;
                case 2:
                        putpixel(x1,y1++,WHITE);
                        putpixel(x1,y1++,BLACK);
                        putpixel(x1,y1++,BLACK);
                        putpixel(x1,y1++,WHITE);
                        break;
                case 3:
                        putpixel(x1,y1++,WHITE);
                        putpixel(x1,y1++,WHITE);
                        putpixel(x1,y1++,BLACK);
                        putpixel(x1,y1++,BLACK);
                  }

}
}
x1++;y1=40;
}
/* generate share2 */
x2=0;y2=100;
//temp1=rand(2);
for(x=0;x<=250;x++)
{
for(y=10;y<=30;y++)
{
if(getpixel(x,y)==WHITE)
{     switch(scheme)
                { case 0:
                        putpixel(x2,y2++,BLACK);
                        putpixel(x2,y2++,WHITE);
                        putpixel(x2,y2++,WHITE);
                        putpixel(x2,y2++,WHITE);
                    //    y2=y2+2;
                        break;
                case 1:
                        putpixel(x2,y2++,WHITE);
                        putpixel(x2,y2++,BLACK);
                        putpixel(x2,y2++,WHITE);
                        putpixel(x2,y2++,WHITE);
```


eISSN 1303-5150

```
                    break;
                case 2:
                    putpixel(x2,y2++,WHITE);
                    putpixel(x2,y2++,WHITE);
                    putpixel(x2,y2++,WHITE);
                    putpixel(x2,y2++,BLACK);
                    break;
                case 3:
                    putpixel(x2,y2++,WHITE);
                    putpixel(x2,y2++,WHITE);
                    putpixel(x2,y2++,BLACK);
                    putpixel(x2,y2++,WHITE);
                  }
}
else
{   switch(scheme)
                  {case 0:
                putpixel(x2,y2++,WHITE);
                putpixel(x2,y2++,BLACK);
                putpixel(x2,y2++,WHITE);
                putpixel(x2,y2++,BLACK);
            //      y2=y2+2;
                break;
                case 1:
                putpixel(x2,y2++,BLACK);
                putpixel(x2,y2++,WHITE);
                putpixel(x2,y2++,BLACK);
                putpixel(x2,y2++,WHITE);
                break;
                case 2:
                putpixel(x2,y2++,WHITE);
                putpixel(x2,y2++,BLACK);
                putpixel(x2,y2++,WHITE);
                putpixel(x2,y2++,BLACK);
                break;
                case 3:
                putpixel(x2,y2++,BLACK);
                putpixel(x2,y2++,WHITE);
                putpixel(x2,y2++,BLACK);
                putpixel(x2,y2++,WHITE);
                break;
                  }
}
}
x2++;y2=100;
}
//x1=0;
//x2=0;
//y1=40;
```


eISSN 1303-5150

```
//y2=70;
x3=0;
y3=200;
//x2=0,y2=70;
//temp1=rand(2);
/*third share*/
for(x=0;x<=250;x++)
{
for(y=10;y<=30;y++)
{
if(getpixel(x,y)==WHITE)
{     switch(scheme)
                    { case 0:
                              putpixel(x3,y3++,BLACK);
                              putpixel(x3,y3++,WHITE);
                              putpixel(x3,y3++,WHITE);
                              putpixel(x3,y3++,WHITE);
                     //       y3=y3+2;
                              break;
                    case 1:
                              putpixel(x3,y3++,WHITE);
                              putpixel(x3,y3++,BLACK);
                              putpixel(x3,y3++,WHITE);
                              putpixel(x3,y3++,WHITE);
                              break;
                    case 2:
                              putpixel(x3,y3++,WHITE);
                              putpixel(x3,y3++,WHITE);
                              putpixel(x3,y3++,WHITE);
                              putpixel(x3,y3++,BLACK);
                              break;
                    case 3:
                              putpixel(x3,y3++,WHITE);
                              putpixel(x3,y3++,WHITE);
                              putpixel(x3,y3++,BLACK);
                              putpixel(x3,y3++,WHITE);
                       }
}
else
{     switch(scheme)
                      {case 0:
                    putpixel(x3,y3++,WHITE);
                    putpixel(x3,y3++,WHITE);
                    putpixel(x3,y3++,BLACK);
                    putpixel(x3,y3++,BLACK);
                     //    y3=y3+2;
                    break;
                    case 1:
                    putpixel(x3,y3++,BLACK);
```


eISSN 1303-5150

```
                putpixel(x3,y3++,WHITE);
                putpixel(x3,y3++,WHITE);
                putpixel(x3,y3++,BLACK);
                break;
                case 2:
                putpixel(x3,y3++,BLACK);
                putpixel(x3,y3++,BLACK);
                putpixel(x3,y3++,WHITE);
                putpixel(x3,y3++,WHITE);
                break;
                case 3:
                putpixel(x3,y3++,WHITE);
                putpixel(x3,y3++,BLACK);
                putpixel(x3,y3++,BLACK);
                putpixel(x3,y3++,WHITE);
                  }
}
}
x3++;y3=200;
}
x4=0;y4=300;
x1=0;y1=40;
x2=0;y2=100;
x3=0;y3=200;
val1=0;val2=0,val3=0;val=0;
flag=3;                                    getch();
/*superimpose three shares  */
for(x=0,x4=0;x<=250;x++,x4=x4+1)
{
for(y=0,y4=300;y<=10;y++,y4=y4+1)
 {
if(getpixel(x1,y1)==WHITE)
 {
   color1=1;
for(i=0;i<4;i++)
 {
   val1+=(pow(2,flag--)*color1);
if(getpixel(x1,y1+i+1)==WHITE)
   color1=1;
else color1=0;
 }
flag=3;
 }
else
 {
 color1=0;
for(i=0;i<4;i++)
 {
   val1+=(pow(2,flag--)*color1);
```


eISSN 1303-5150

```
if(getpixel(x1,y1+i+1)==WHITE)
   color1=1;
else color1=0;
 }
flag=3;
 }
if(getpixel(x2,y2)==WHITE)
 {
 color2=1;
for(i=0;i<4;i++)
 {
   val2+=(pow(2,flag--)*color2);
if(getpixel(x2,y2+i+1)==WHITE)
   color2=1;
else color2=0;
 }
flag=3;
 }
else
 {
 color2=0;
for(i=0;i<4;i++)
 {
   val2+=(pow(2,flag--)*color2);
if(getpixel(x2,y2+i+1)==WHITE)
   color2=1;
else color2=0;
 }
flag=3;
 }
if(getpixel(x3,y3)==WHITE)
 {
 color3=1;
for(i=0;i<4;i++)
 {
   val3+=(pow(2,flag--)*color3);
if(getpixel(x3,y3+i+1)==WHITE)
   color3=1;
else color3=0;
 }
flag=3;
 }
else
 {
 color3=0;
for(i=0;i<4;i++)
 {
   val3+=(pow(2,flag--)*color3);
if(getpixel(x3,y3+i+1)==WHITE)
```

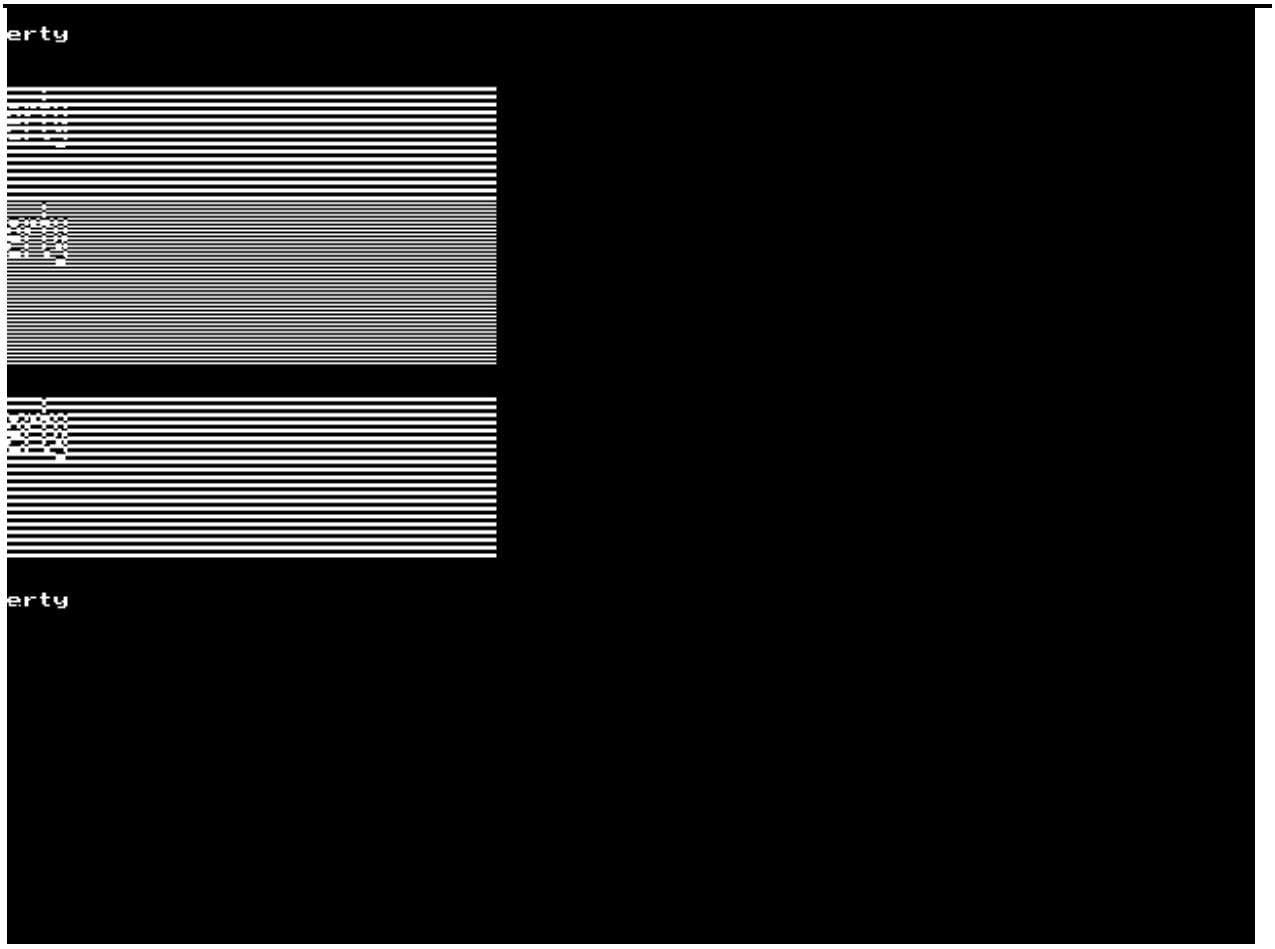

eISSN 1303-5150

```
   color3=1;
else color3=0;
 }
flag=3;
 }
color=0;
val=val1^val2;
for(i=0;i<4;i++)
 {
 color+=(pow(2,i)*(1-(val%2)));
val/=2;
 }
val=color;
color=0;
//  temp;
count=0;
 // save=val;
for(i=0;i<4;i++)
 {
save=val;
color=val%2;
temp=val3%2;
val=1-(color^temp);
if(val)
count+=1;
val=save;
val=val/2;
 val3/=2;
 }

if(count>2)
 {
putpixel(x4,y4,WHITE);
//  y3=y3+2;
//  putpixel(x4+1,y4,WHITE);
//  putpixel(x4,y4+1,WHITE);
 // putpixel(x4+1,y4+1,WHITE);
 }
else
 {
putpixel(x4,y4,BLACK);
 // y3=y3+2;
//  putpixel(x4+1,y4,BLACK);
//  putpixel(x4,y4+1,BLACK);
//  putpixel(x4+1,y4+1,BLACK);
 }
 y1=y1+4;
 y2=y2+4;
 y3=y3+4;
```


eISSN 1303-5150

```
}
x1=x1+1;
x2=x2+1;
  y1=40;
  y2=100;
x3=x3+1;
  y3=200;
flag=3;
 // x4+=2;
 // y4=300;
 val1=val2=val3=val=0;
}

getch();
restorecrtmode();
closegraph();
}
```


OUTPUT:



eISSN 1303-5150

REFERENCES:

[1] M. Naor and A. Shamir, Visual cryptography, in Advances in Cryptology(Lecture Notes in Computer Science), vol. 950. Berlin, Germany:Springer-Verlag, 1995.

[2] Thomas Monoth, BabuAnto P. ICEBT 2010.Procedia Computer Science 2 (2010) 143–148.

[3] SowmyaSuryadevara, RohailaNaaz, Shweta, ShuchitaKapoor, AnandSharmaVisual Cryptography Improvises the Security of Tongue as a Biometric in Banking SystemComputer Science Dept.Mody Institute of Technology International Conference on Computer & Communication Technology (ICCCT)-2011

[4]Modified3-by-4vcs scheme for grayscaleimagesusing 4 subpixel layouts: Extension of 2-by-2 vcsscheme using 2 subpixellayouts:ISSN: 2096-3246 Volume 54, Issue 02, July, 2022