J. Inst. Eng. India Ser. B (October 2020) 101(5):609–621
https://doi.org/10.1007/s40031-020-00475-4

ORIGINAL CONTRIBUTION

Polynomial-Based Secret Sharing Scheme for Text, Image and Audio

Prashanti Guttikonda[1,2] · **Nirupama Bhat Mundukur**[1]

Received: 14 October 2019 / Accepted: 26 July 2020 / Published online: 7 August 2020

Abstract Confidential information can be shared in a group with secret sharing scheme where no single user can retrieve the secret. Only few or more in the group together only can restore the secret. The secret could be in the form of numeric, text, image, audio, etc. There are different methods to deal with the above said secrets. In this paper, Lagrange's interpolation method is applied in the formation and restoration of shares on numeric, text, image and audio. Constant coefficients and random coefficients are used while considering the coefficients of polynomial and dissimilarity between shares, and secret is measured using Pearson correlation coefficient.

Keywords Secret sharing · Text · Image · Audio · Polynomial · Lagrange's interpolation · Pearson correlation

Introduction

Cryptography and watermarking algorithms are not suitable for situations where secret or master key has to be distributed and controlled among a group of participants [1]. For such cases, a branch of cryptography known as secret sharing scheme can be applied where the secret is cleaved into meaningless shares and allocated to the members of the group. Only a valid subset of members in the group can reveal the secret by combining their shares [2]. With huge expansion of network, data communication is not limited to only text, but it has expanded to handle image, audio, video, etc. So there is a need for a scheme that can provide security to all these types of multimedia data.

A threshold scheme based on polynomial interpolation has been introduced by Shamir that works well with text, image and audio based on constraints of the application and requirements of users [3]. In this method for a secret integer value u, a group of n participants with a threshold value k generate n shares with a polynomial of $k-1$ degree. The secret can be reconstructed by a group with at least k participants using Lagrange's interpolation, and the group with less than k participants cannot obtain the information. When secret is in the form of image or video, then Shamir's scheme generates shares with size equal to the original image which is a burden for storage space [4]. To overcome this, Thien and Lin suggested a procedure where polynomial considers the pixel values as the coefficient thereby minimizing the dimensions of the image shares [5]. Security of shares is enhanced with the scheme proposed by Lin and Tsai, where the shares are hidden into camouflage images and then delivered to the participant, and also provides the capability of authentication during recovery process. Wang and Shyu proposed a scheme where the secret image is reconstructed in a scalable manner depending on priorities of the shares presented by the participant [6].

Similar to images, confidential audios can be encrypted into n shares and simultaneous playing of at least k shares reveals the secret audio. Yvo et al. proposed ASS scheme which uses sound inference property to embed messages into music, and this scheme has been limited to $(2, n)$ threshold [7]. Daniel and Spyros proposed (k, n) audio visual cryptography scheme using the existing general

✉ Prashanti Guttikonda
prashantiguttikonda77@gmail.com

[1] Vignan's Foundation for Science, Technology and Research, Vadlamudi, Guntur District, Andhra Pradesh, India

[2] Vignan's LARA Institute of Technology and Science, Vadlamudi, Guntur District, Andhra Pradesh, India

access structure in visual cryptography scheme [8]. Huan et al. embedded the n shares generated into n shelter audios thereby improving the security of the ASS scheme [9, 10].

Proposed Method

In the proposed method, polynomial-based approach is used for sharing the secret where the secret is text, image and audio. As polynomial is defined by coefficients, shares can be generated either with constant coefficients or with random coefficients. With constant coefficients, the coefficients of the polynomial will be same for the entire secret, whereas with random, different coefficients are taken randomly for each element in the secret. To restore the secret, we have to rebuild the polynomial as

$$f(x) = S_1 \frac{(x-x_2)(x-x_3)\ldots(x-x_k)}{(x_1-x_2)(x_1-x_3)\ldots(x_1-x_k)} + S_2 \frac{(x-x_1)(x-x_3)\ldots(x-x_k)}{(x_2-x_1)(x_2-x_3)\ldots(x_2-x_k)} + \cdots + S_k \frac{(x-x_1)(x-x_2)\ldots(x-x_{k-1})}{(x_k-x_1)(x_k-x_2)\ldots x_k-x_{k-1})} \bmod q \quad (1)$$

And the secret is acquired as

$$\begin{aligned} f(x=0) &= S_1 \times (0-x_2)(0-x_3)\ldots(0-x_k) \\ &\times \text{modulo inverse}\,((x_1-x_2)(x_1-x_3)\ldots(x_1-x_k), q) \\ &+ S_2 \times (0-x_1)(0-x_3)\ldots(0-x_k) \\ &\times \text{modulo inverse}\,((x_2-x_1)(x_2-x_3)\ldots(x_2-x_k), q) + \cdots \\ &+ S_k \times (0-x_1)(0-x_2)\ldots(0-x_{k-1}) \\ &\times \text{modulo inverse}\,((x_k-x_1)(x_k-x_2)\ldots(x_k-x_{k-1}), q) \bmod q \end{aligned} \quad (2)$$

Text sharing scheme for (k, n) threshold

Algorithm 1: Text Share Generation

Input: Secret text $T = \{L_1, L_2, L_3 \ldots L_m\}$ where L_m is the letter in the text at the mth position.
Output: secret shares $S1$, $S2$, $S3 \ldots S_n$
Step 1: The secret text (T) to be shared is in the form of characters. It has to be transformed into integer values. The characters of text are mapped to their corresponding ASCII values. Now, the secret text is in the form of integers (T') which are used for secret generation.
Step 2: Calculate maximum of T' and a number (q) which is first prime and greater than this maximum.
Step 3: Choose coefficients a_0 as secret value and a_1, a_2,…a_{k-1} randomly from GF(q).
Step 4: Generate arrays $m_1[]$, $m_2[]$,…$m_{k-1}[]$ with random numbers equal to the length of the text which defines the coefficients of the polynomial equation.

Step 5: With constant coefficients, generate share for xth participant such that $x \in [1, n]$, using equation

$$S_x = T'(i) + a_1 \times x^1 + a_2 \times x^2 + a_3 \times x^3 + \cdots + a_{k-1} \times x^{k-1} \bmod q$$

where $T'(i)$ is the ith secret value in T'

$a_1, a_2, \ldots a_{k-1}$ are values obtained in step 3

Step 6: With random coefficients, generate share for xth participant such that $x[1, n]$, using equation

$$S_x = T'(i) + m_1[i] \times x^1 + m_2[i] \times x^2 + m_3[i] \times x^3 + \cdots + m_{k-1}[i] \times x^{k-1} \bmod q$$

where $T'(i)$ is the ith secret value in T'

$m_1[], m_2[], \ldots m_{k-1}[]$ are values obtained in step 4

Step 7: As the share generated in step 5 or step 6 is integers, they are to be converted into text. The integers are matched against ASCII table, and the corresponding characters are extracted. These characters are combined to get shares in the form of text.
Step 8: If the shares are to be generated with constant coefficients, then

for x = 1 to n % number of shares
For i = 1 to length (T')
Do steps 5 and 7
End for
return S_x
End for

Step 9: For generation of shares with random coefficients, do

for x = 1 to n % number of shares
For i = 1 to length (T')
Do steps 6 and 7
End for
return S_x
End for

Step 10: Finally, the pair (x, S_x) is provided to the x participant where $x \in [1, n]$.

Algorithm 2: Text Reconstruction

Input: Secret shares S_1, S_2, $S_3 \ldots S_n$
Output: Secret text $T = \{L_1, L_2, L_3 \ldots L_m\}$
Description
Step 1: The participants who want to reveal the secret must be more than k and should submit their shares as (x_i, S_i). As the shares are in the form of text, they are translated into integers using ASCII table, and then, a polynomial of degree $k - 1$ can be rebuilt using (1) and the secret T' is obtained with $f(x = 0)$ using (2).
Step 2: As T' is our secret which is in the form of integers, mapping these to ASCII table and taking the corresponding characters we get the secret text T.

Results of Text Sharing Process

Apply $k = 2$ and $n = 3$ for (k, n) threshold scheme for a secret text, which means at least 2 shares are required to get the secret. Figure 1a shows the shares generated by constant coefficients with a polynomial of degree $(k - 1)$ which will be 1, since we have considered $k = 2$.

Our polynomial to generate shares is

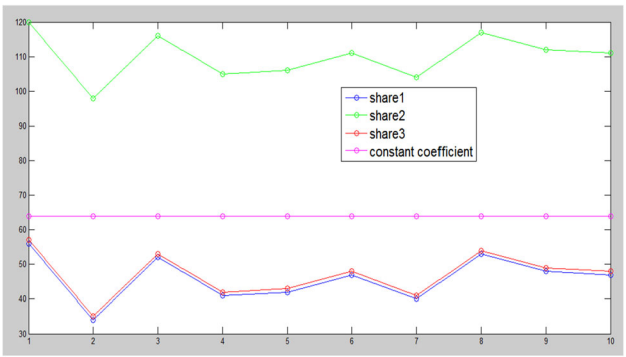
$$S_x = (a_0 + 64 \times x) \bmod q$$

```
original text: washington
share 1    8"4)*/(50/

share 2    xbtijohupo

share 3    9#5*+0)610
combining share 1 and share 2   washington
```

(a)



(b)

Fig. 1 **a** Shares generated with constant coefficients; **b** graph for shares generated with constant coefficient '64'

where a_0 is the secret and 64 is the constant coefficient for $x = 1, 2, 3$ $(n = 3)$. From the graph in Fig. 1b, we can observe that shares generated are of same pattern and same characters in the text are replaced with same cipher text character as shown in Table 1. An analyst observing the shares can make assumption of the secret text.

Figure 2a shows the shares generated with the polynomial of the form

$$S_x = (a_0 + r \times x) \bmod q$$

where r takes different random numbers for each character a_0 of the text. These random numbers act as a blending factor and generate shares of different patterns shown in Fig. 2b thereby optimizing the security of text when compared with constant coefficients.

The details in Table 2 suggest that with random coefficient each character of the plaintext is replaced with different cipher text characters, and analyst finds difficult to make assumption of the secret text. Empty cells in table represent 'space' character, that is, characters t, o, n are replaced with space character.

Secret image sharing.

Algorithm 3: Image Share Generation

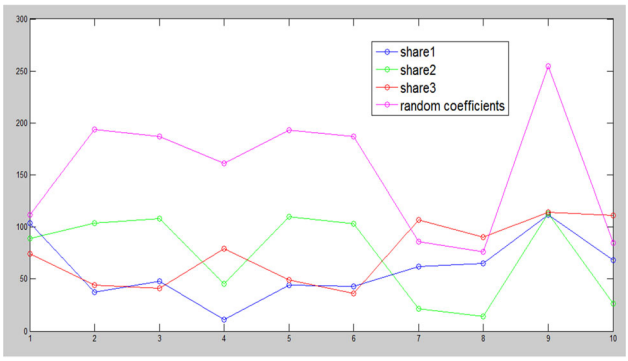
Input: Secret image $I = \{P_1, P_2, P_3 \ldots P_m\}$ where P_m is the pixel value in the image at the mth position.
Output: Secret shares S_1, S_2, $S_3 \ldots S_n$
Step 1: Get image file (I) with imread() function.

```
original text: washington
share 1    h%0,+>ApD

share 2    Yhl-ngq

share 3    J,)O1$kZro
combining share 1 and share 2   washington
```

(a)



(b)

Fig. 2 **a** Shares generated with random coefficients; **b** graph for shares generated with random coefficient

Table 1 Shares generated for text with constant coefficients

Plain text	w	a	s	h	i	n	g	t	o	n
Share1	8	“	4	)	*	/	(	5	0	/
Share2	x	b	t	i	j	o	h	u	p	o
Share3	9	#	5	*	+	0	)	6	1	0

Table 2 Shares generated for text with random coefficients

Plain text	w	a	s	h	i	n	g	t	o	n
Share1	h	%	0	,	+	>	A	p	D	
Share2	Y	h	l	–	n	g	q			
Share3	J	,	)	O	l	$	k	Z	r	o

Step 2: Calculate the maximum of $\{P_1, P_2, P_3 \ldots P_m\}$ and consider a number (q) which is first prime and greater than this maximum.
Step 3: Each value of $\{P_1, P_2, P_3 \ldots P_m\}$ is a secret and considered as a_0.
Step 4: Choose coefficients $a_1, a_2, \ldots a_{k-1}$ randomly from GF(q).
Step 5: Generate arrays $m_1[], m_2[], \ldots m_{k-1}[]$ with random numbers equal to the length of the image which defines the coefficients of the polynomial equation.
Step 6: With constant coefficients, generate share for xth participant such that $x \in [1, n]$, using equation

$$S_x = I(i) + a_1 \times x^1 + a_2 \times x^2 + a_3 \times x^3 + \cdots + a_{k-1} \times x^{k-1} \bmod q$$

$I(i)$ is the ith value in $\{P_1, P_2, P_3 \ldots P_m\}$

$a_1, a_2, \ldots a_{k-1}$ are values obtained in step 4

Step 7: With random coefficients, generate share for xth participant such that $x \in [1, n]$, using equation

$$S_x = I(i) + m_1[i] \times x^1 + m_2[i] \times x^2 + m_3[i] \times x^3 + \cdots + m_{k-1}[i] \times x^{k-1} \bmod q$$

$I(i)$ is the ith value in $\{P_1, P_2, P_3 \ldots P_m\}$

$m_1[], m_2[], \ldots m_{k-1}[]$ are values obtained in step 5

Step 8: If the shares are to be generated with constant coefficients, then

```
for x = 1 to n % number of shares
    For i = 1 to size of image
        Do step 6
    End for
    return S_x
End for
```

Step 9: For generation of shares with random coefficients, do

```
for x = 1 to n % number of shares
    For i = 1 to size of image
        Do step 7
    End for
    return S_x
End for
```

Step 10: Finally, the pair (x, S_x) is provided to the xth participant where $x \in [1, n]$.

Algorithm 4: Image Reconstruction

Input: Secret shares $S_1, S_2, S_3 \ldots S_n$
Output: Secret image $I = \{P_1, P_2, P_3 \ldots P_m\}$
Step 1: At least k participants should present their shares (x, S_x) to reveal the secret, and with them, a polynomial

of degree $k - 1$ can be rebuilt using (1) and the secret I is obtained with $f(x = 0)$ using (2).
Step 2: Return I (original image)

Results of Image Sharing Process

For Grayscale:

The results in Fig. 3 are for polynomial equation of the form

$$S_x = (I + 64 \times x) \bmod q$$

where I is the secret image and 64 is the constant coefficient. For x = 1,2,3, three shares are generated. Since it is a (2, 3) threshold scheme, any of 2 or 3 shares are required to reconstruct the secret. Here we combined 1 and 2 shares to get the original image. Figure 4 shows a plot for first 1000 pixels; values of share1, share2 and share3 show that all are in same pattern and not much scrambling is done on pixel values. Results and graph shows that the shares are not totally encrypted and are revealing the secret.

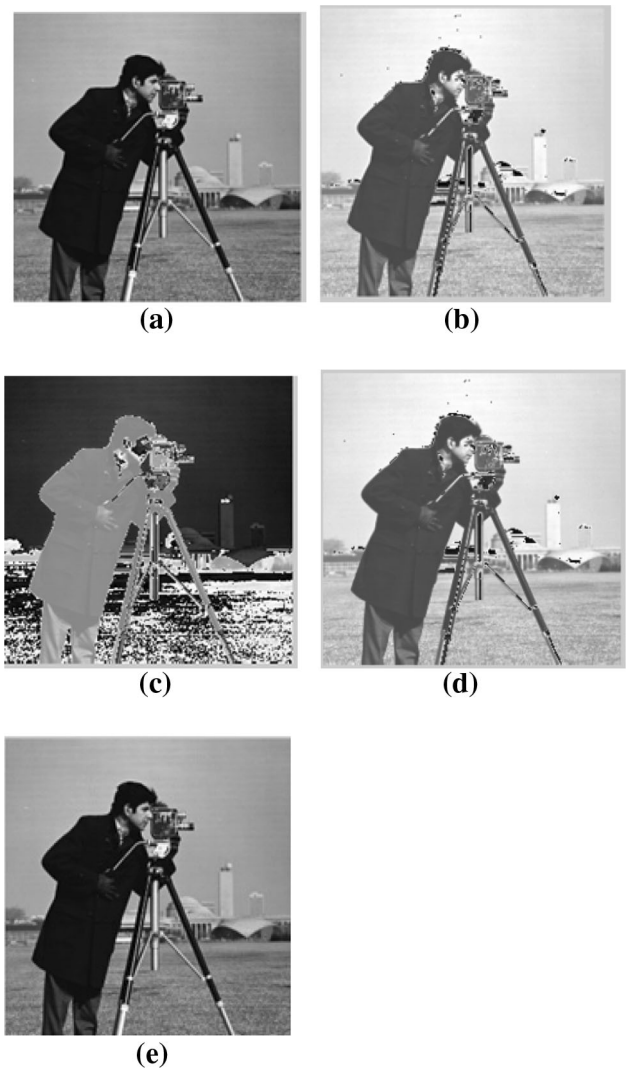


Fig. 3 **a** Original image; **b**, **c** and **d** shares generated for participants 1, 2 and 3 with constant coefficients; **e** reconstructed image after combining share 1 and share 2

With constant coefficient, each pixel value is encrypted with a constant '64,' whereas random coefficient takes different values for different pixel values. For this, we generate random numbers(r) equal to size of image and apply it to the polynomial

$$S_x = (I(i) + r(i) \times x) \bmod q$$

Figure 5 shows results of applying above polynomial, and Fig. 6 shows graph for first 1000 pixel values of three shares. Results and graph show that the shares are entirely scrambled giving no information about the secret image thereby enhancing the security of the secret in contrast to constant coefficient.

For color images

Figures 7 and 8 show results of color images. The R, G, B components are extracted from color image. Each individual component goes through the Algorithm 2 described for image share generation. Finally, R, G, B components are concatenated to get the shares.

From Fig. 7, it can be observed that encrypting the image with constant coefficient reveals the secret. Using random coefficient distorts the entire image thereby maintain privacy as demonstrated in Fig. 8.

Audio secret sharing

Algorithm 5: Audio Share Generation

Input: Secret audio $A = \{A_1, A_2, A_3 \ldots A_m\}$ where A_m is the amplitude value in the audio at the mth time interval.
Output: Secret shares $A_1, A_2, A_3 \ldots A_n$
Step 1: Read the secret audio A.
Step 2: A = round($A \times 10^d$), where d is some positive integer.

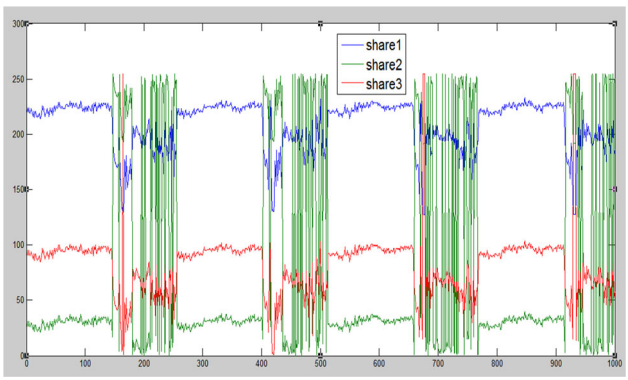


Fig. 4 Graph for shares generated with constant coefficient '64'

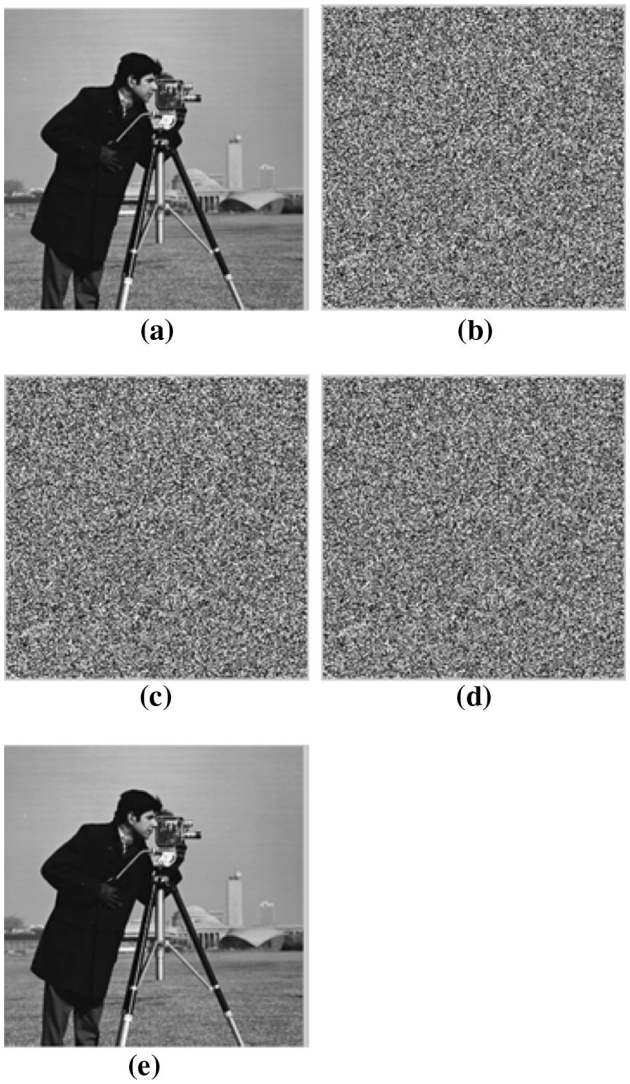


Fig. 5 **a** Native image; **b**, **c** and **d** are shares generated for participants 1, 2 and 3 with random coefficients; **e** reconstructed image after combining share 1 and share 2

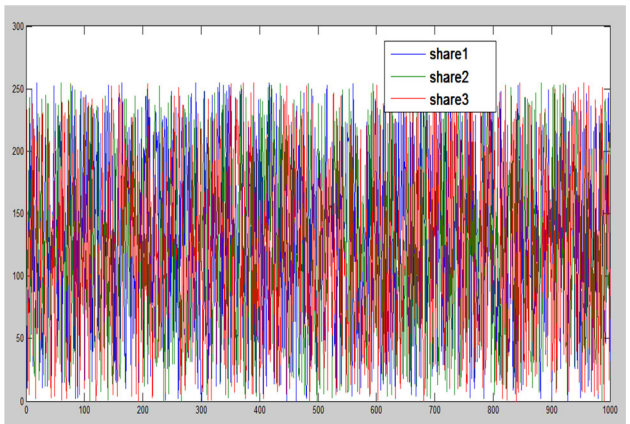


Fig. 6 Graph for shares generated with random coefficient

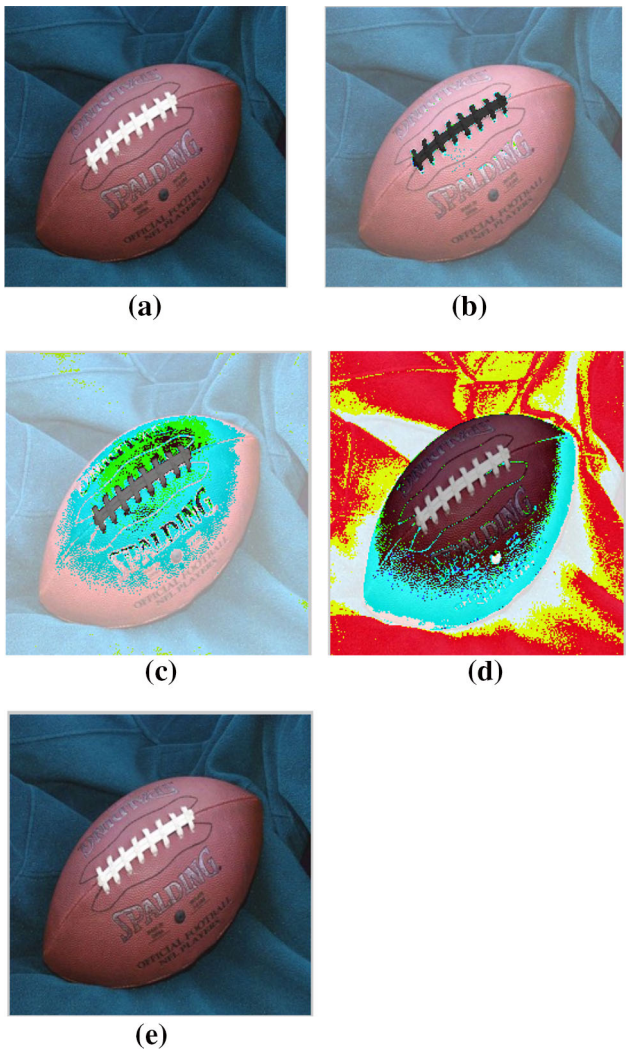


Fig. 7 **a** Native image; **b**, **c** and **d** are shares generated for participants 1, 2 and 3 with random coefficients; **e** reconstructed image after combining share 1 and share 2

Step 3: Determine the minimum value (m) of A and do $m' = \text{abs}(m)$ to make it positive integer.
Step 4: $A' = A + m'$
Step 5: Calculate the maximum amplitude value in A' and consider the first prime number (q) which is greater than this maximum.
Step 6: Each amplitude value of the audio A' is a secret and considered as a_0.
Step 7: Choose coefficients $a_1, a_2, \ldots a_{k-1}$ randomly from GF(q).
Step 8: Generate arrays $m_1[], m_2[], \ldots m_{k-1}[]$ with random numbers equal to the length of the audio which defines the coefficients of the polynomial equation.
Step 9: With constant coefficients, generate share for xth participant such that $x \in [1, n]$, using equation

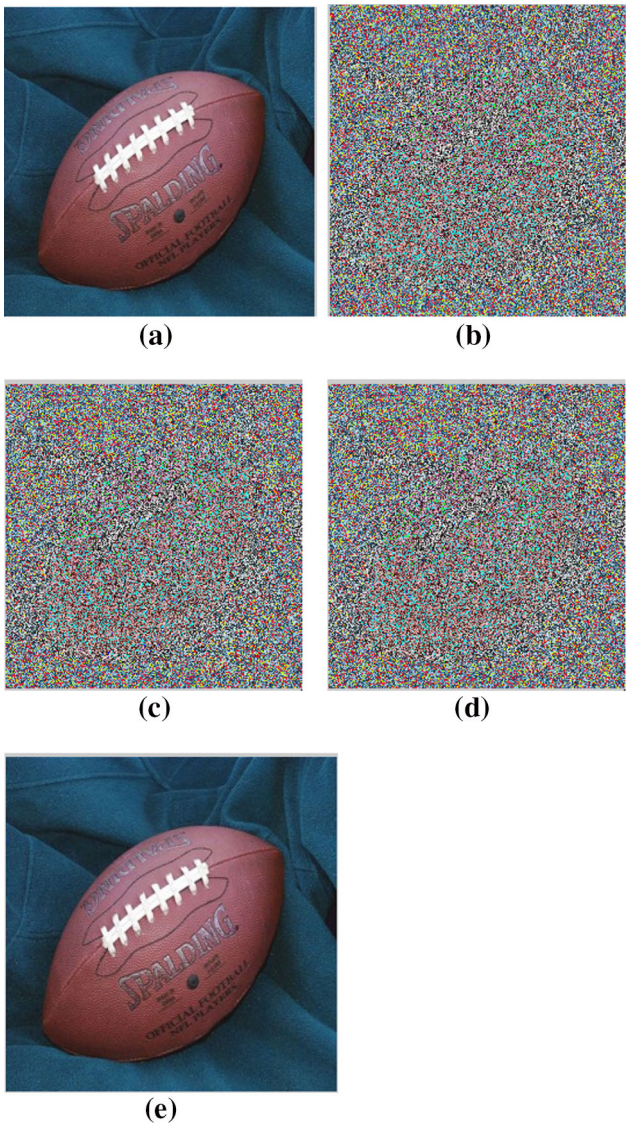


Fig. 8 **a** Native image; **b**, **c** and **d** are shares generated for participants 1, 2 and 3 with random coefficients; **e** reconstructed image after combining share 1 and share 2

$$S_x = A'(i) + a_1 \times x^1 + a_2 \times x^2 + a_3 \times x^3 + \cdots + a_{k-1} \times x^{k-1} \bmod q$$

where $A'(i)$ is the ith value in A'

$a_1, a_2, \ldots a_{k-1}$ are values obtained in step 4

Step 10: With random coefficients, generate share for xth participant such that $x \in [1, n]$, using equation

$$S_x = A'(i) + m_1[i] \times x^1 + m_2[i] \times x^2 + m_3[i] \times x^3 + \cdots + m_{k-1}[i] \times x^{k-1} \bmod q$$

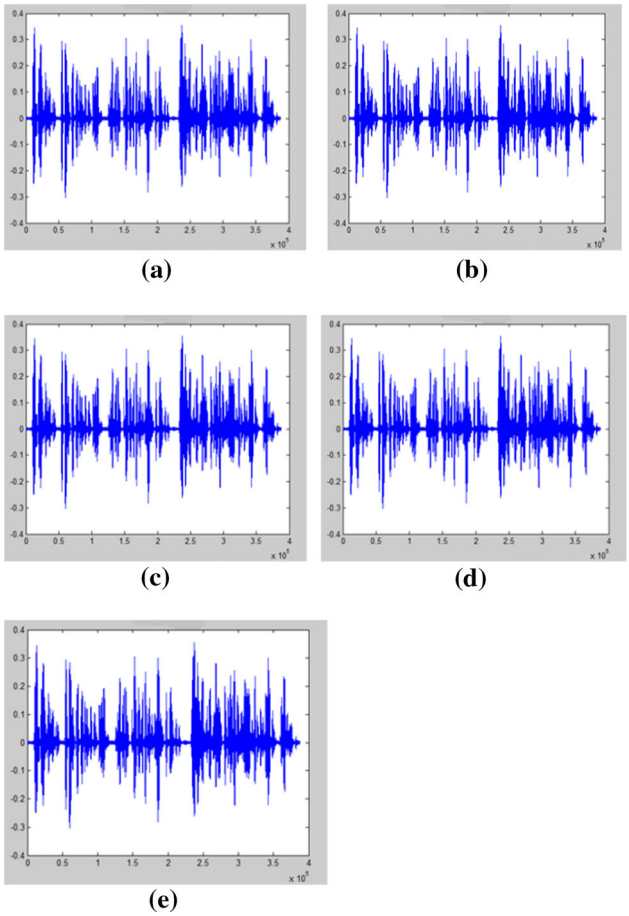


Fig. 9 **a** Original audio; **b** through **d** shares generated for participants 1 through 3 with constant coefficients; **e** reconstructed audio after combining share 1 and share 2

where $A'(i)$ is the ith value in A'

$m_1[\,], m_2[\,], \ldots m_{k-1}[\,]$ are values obtained in step 4

Step 11: If the shares are to be generated with constant coefficients, then

for $x = 1$ to n % number of shares
For $i = 1$ to length of audio
Do step 9
End for
return S_x
End for

Step 12: If the shares are to be generated with random coefficients, then

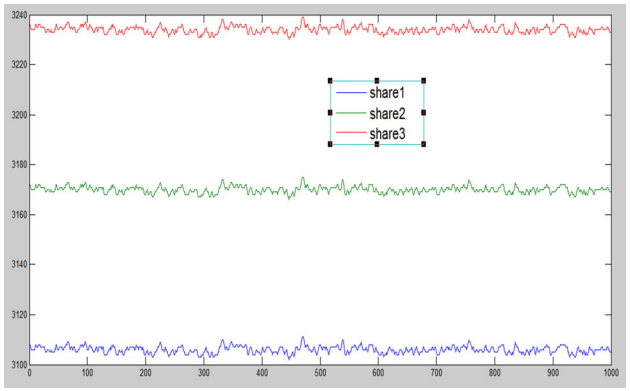


Fig. 10 Graph for audio shares generated with constant coefficient '64'

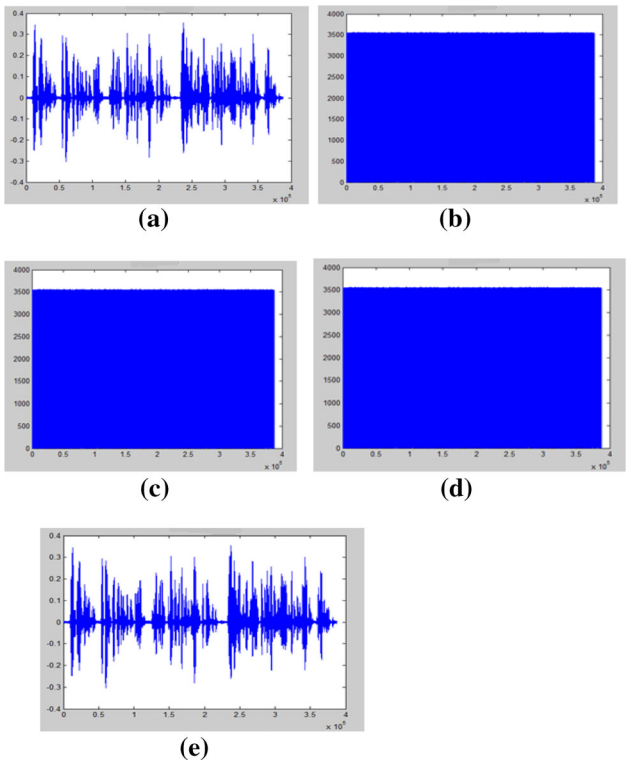


Fig. 11 **a** Native audio; **b** through **d** shares generated for participants 1 through 3 with random coefficients; **e** reconstructed audio after combining share 1 and share 2

for $x = 1$ to n % number of shares
  For $i = 1$ to length of audio
    Do step 10
  End for
  return S_x
End for

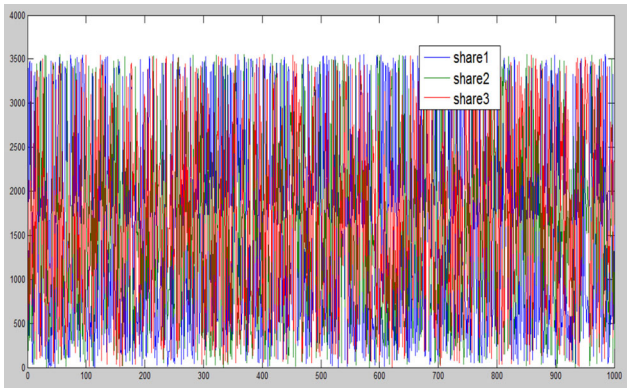


Fig. 12 Graph for audio shares generated with random coefficient

Step 13: Finally, the pair (x, S_x) is provided to the xth participant where $x \in [1, n]$.

Algorithm 6: Audio Reconstruction

Input: Secret shares $S_1, S_2, S_3 \ldots S_n$
Output: Secret audio $A = \{A_1, A_2, A_3 \ldots A_m\}$
Step 1: At least k participants should present their shares (x, S_x) to reveal the secret, and with them, a polynomial of $k - 1$ degree can be rebuilt using (1) and the secret A' is obtained with $f(x = 0)$ using (2).
Step 2: $A = (A' - m')/10^d$
where d and m' are obtained from Algorithm 5, steps 2 and 3.
Step 3: Return A.

Results of Audio Sharing Process

For $k = 3$ and $n = 3$, we generate a polynomial of degree 1 which is of the form

$$S_x = (A' + 64 \times x) \bmod q$$

Applying the above equation to preprocessed audio A' and with constant coefficient 64 with $x = 1,2,3$, generate shares shown in Fig. 9.

Plot of first 1000 amplitude values for three shares is shown in Fig. 10. As all amplitude values are processed with same value '64,' we can observe that audio shares generated are in same fashion with the original audio and so the secret is clearly audible.

Figure 11 shows results of random coefficients with equation

$$S_x = (A'(i) + r(i) \times x) \bmod q$$

The graph for first 1000 amplitude values of three shares generated with above polynomial is shown in Fig. 12.

Table 3 Average processing time for text with constant coefficient

Test file	Length of text (characters)	Share creation (s)	Secret reconstruction (s)
Text1	10	0.0012	0.00346
Text2	374	0.0021	0.0014
Text3	3391	0.00986	0.00496
Text4	26,382	0.0569	0.02620
Text5	70,352	0.17896	0.07243
Text6	246,232	0.57283	0.3420
Text7	378,142	0.9341	0.32483

Table 4 Average processing time for text with random coefficient

Test file	Length of text (characters)	Share creation (s)	Secret reconstruction (s)
Text1	10	0.004854	0.0037945
Text2	374	0.0060	0.00496
Text3	3391	0.01516	0.00533
Text4	26,382	0.079933	0.04916
Text5	70,352	0.381	0.24123
Text6	246,232	2.1595	0.57085
Text7	378,144	3.4355	0.63653

Since amplitude values are processed with different values of r instead of single value as in constant coefficient, the shares generated are totally noisy and not audible.

Analysis of Results

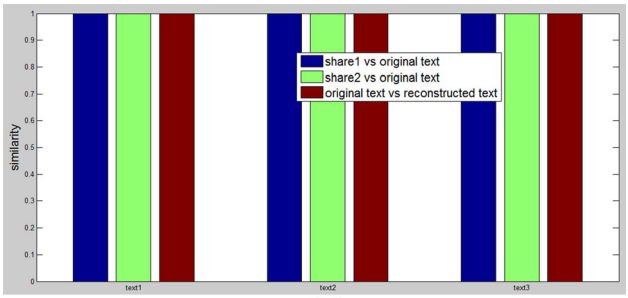
We apply $k = 2$ and $n = 3$ for (k, n) threshold scheme, which means at least 2 shares are required to get the secret. The proposed method is implemented using MATLAB and with different samples of secret data.

Analysis of Results on Text Sharing Process

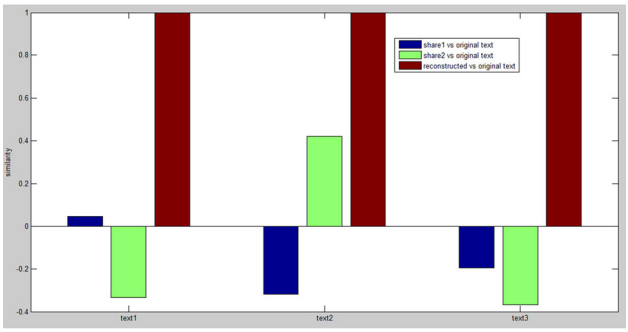
The details of processing time to create three shares and to reconstruct the secret with two shares for polynomial with constant and random coefficient are given in Tables 3 and 4.

Since random numbers equal to the size of text are to be generated in random coefficient, it takes more processing time than the constant coefficient for generating shares. Both tables suggest that time required for reconstructing the secret is less than creating the shares.

Figure 13 illustrates similarity among native text, its shares and the restored share. Similarity values are shown in Tables 5 and 6. In Fig. 13a, three different texts are



(a)



(b)

Fig. 13 **a** Similarity plot for text with constant coefficients; **b** Similarity plot for text with random coefficients

Table 5 Similarity score for text with constant coefficient

Sample text	Share1	Share2	Reconstructed text
Text 1	1.0000	1.0000	1.0000
Text 2	1.0000	1.0000	1.0000
Text 3	1.0000	1.0000	1.0000

taken and followed Algorithm-1 described for secret sharing process with constant coefficients using equation

$$S_x = T'(i) + a_1 \times x^1 + a_2 \times x^2 + a_3 \times x^3 + \cdots + a_{k-1} \times x^{k-1} \bmod q$$

The corollary shows 1% correlation which means there is close association between them. Thus, there exists content similarity between the shares and the original secret which is not a desirable one.

Likewise Fig. 13b shows results of secret generation with random coefficients by applying equation

$$S_x = T'(i) + m_1[i] \times x^1 + m_2[i] \times x^2 + m_3[i] \times x^3 + \cdots + m_{k-1}[i] \times x^{k-1} \bmod q$$

The results exhibit smaller amount of similarity among original text and its shares. Thus, each share is distinct and provides no information about the secret. We can also

observe that there is 100% correlation between the reconstructed and original texts which implies there is negligible loss of information.

Analysis of Results on Image Sharing Process

Grayscale Image

Images of different dimensions are processed with constant and random coefficients, and their processing times are listed in Tables 7 and 8. It shows that participants can reconstruct the secret in less time when compared to share creation.

Similarity values of image in Table 9 have negative values nearer to − 1 which shows that the shares have inverse correlation with the original secret, whereas positive values nearer to 1 indicate there is positive correlation. The values in Table 10 are nearly equal to 0 which shows there is no similarity between shares and original image.

The correlation among the native image, its shares and regenerated image is shown in Fig. 14. In Fig. 14a, we can observe that the correlation between the native image and its shares is nearly 1% which means that they are highly correlated and the secret is visible in shares as shown in Fig. 4, whereas for random coefficients in Fig. 14b the correlation is around 0% which suggests there is dissimilarity among the original image and its shares. This dissimilarity is shown in Fig. 5. It is also evident that correlation between actual image and the reconstructed image is 100% that indicates minimal loss of information.

Color Images

The average processing time for different color images is shown in Tables 11 and 12. Due to three sample representation of color images, the processing time is more when compared to grayscale images that have single sample representation for each pixel.

Tables 13 and 14 give similarity values for color images. Positive and negative values nearer to 1 or − 1 indicate that there exists similarity between shares and original image. The values that are nearer to zero indicate that there is no similarity between the images.

Figure 15 shows the plot for these tables. We can observe that similarity values between share and original image are nearly zero with random coefficients that shows dissimilarity, whereas with constant coefficients it is either positive value or negative value which means they are very much similar. The similarity between original and reconstructed images is 1 which means that both are exactly same.

Table 6 Similarity score for text with random coefficient

Sample text	Share1	Share2	Reconstructed text
Text 1	0.0466	− 0.3338	1.0000
Text 2	− 0.3188	0.4204	1.0000
Text 3	− 0.1973	− 0.3666	1.0000

Table 7 Average processing time for grayscale with constant coefficient

Test file	Image size	Share creation (s)	Secret reconstruction (s)
Image1	256 × 256	0.44634	0.16734
Image2	291 × 240	0.57602	0.15906
Image3	512 × 512	0.61803	0.20553
Image4	1024 × 1024	0.9097	0.29386
Image5	1620 × 1024	1.0653	0.3146

Table 8 Average processing time for grayscale with random coefficient

Test file	Image size	Share creation (s)	Secret reconstruction (s)
Image1	256 × 256	0.552975	0.1675
Image2	291 × 240	0.6653	0.2013
Image3	512 × 512	0.70233	0.220
Image4	1024 × 1024	1.0324	0.304866
Image5	1620 × 1024	1.5164	0.35305

Table 9 Similarity score for images with constant coefficients

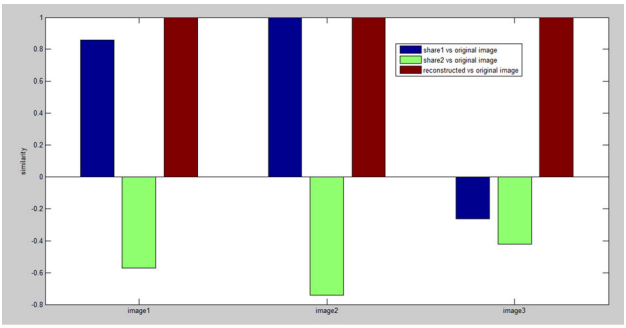
Sample image file	Share1	Share2	Reconstructed image
Image 1	0.8576	− 0.5724	1.0000
Image 2	1.0000	− 0.7404	1.0000
Image 3	− 0.2649	− 0.4222	1.0000

Table 10 Similarity score for images with random coefficients

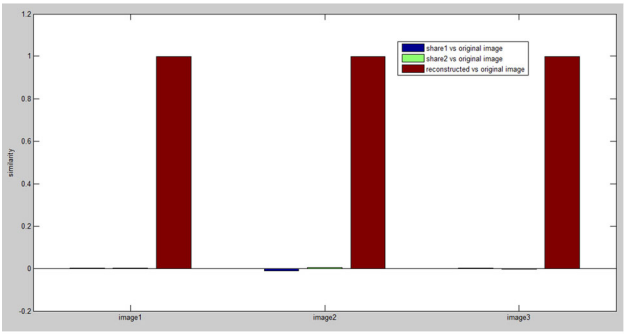
Sample image file	Share1	Share2	Reconstructed image
Image 1	0.0022	4.0348e−04	1.0000
Image 2	− 0.0110	0.0031	1.0000
Image 3	0.0026	− 0.0050	1.0000

Analysis of Results on Audio Sharing Process

Processing time for audio share creation and reconstruction with constant and random coefficients is shown in Tables 15 and 16. Both tables suggest that time for



(a)



(b)

Fig. 14 **a** Similarity plot for images with constant coefficients; **b** similarity plot for images with random coefficients

Table 11 Average processing time for color images with constant coefficient

Test file	Image size	Share creation (s)	Secret reconstruction (s)
Image1	194 × 259 × 3	0.53006	0.1782
Image2	168 × 300 × 3	0.5656	0.18874
Image3	256 × 256 × 3	0.9126	0.4282
Image4	850 × 569 × 3	1.0038	0.4440
Image5	768 × 1024 × 3	1.177633	0.581
Image6	1680 × 1680 × 3	2.22838	1.3412

Table 12 Average processing time for color images with random coefficient

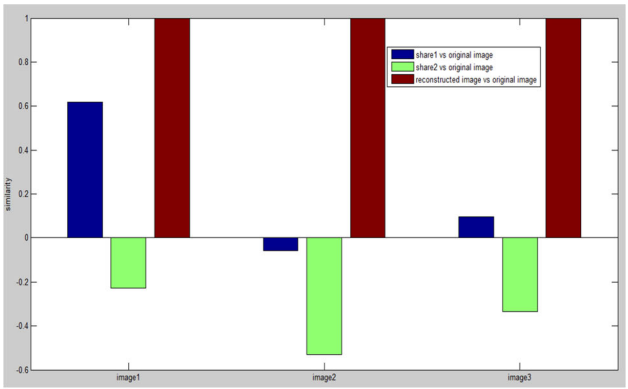
Test file	Image size	Share creation (sec)	Secret reconstruction (sec)
Image1	194 × 259 × 3	0.5887	0.1929
Image2	168 × 300 × 3	0.5441	0.2023
Image3	256 × 256 × 3	0.5537	0.19756
Image4	850 × 569 × 3	0.9888	0.4416
Image5	768 × 1024 × 3	1.29953	0.6398
Image6	1680 × 1680 × 3	2.9409	1.6685

Table 13 Similarity score for color images with constant coefficients

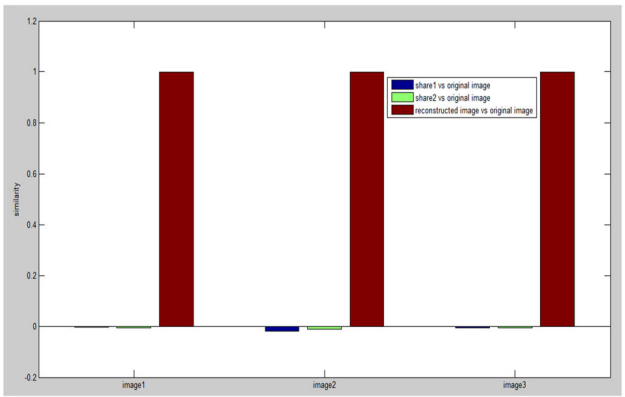
Sample image file	Share1	Share2	Reconstructed image
Color image1	0.6187	− 0.2282	1.0000
Color image2	− 0.0574	− 0.5316	1.0000
Color image3	0.0948	− 0.3335	1.0000

Table 14 Similarity score for color images with random coefficients

Sample image file	Share1	Share2	Reconstructed image
Color image1	− 0.0045	− 0.0057	1.0000
Color image2	− 0.0198	− 0.0111	1.0000
Color image3	− 0.0077	− 0.0074	1.0000



(a)



(b)

Fig. 15 **a** Similarity plot for color images with constant coefficients; **b** similarity plot for color images with random coefficients

constructing and reconstructing the secret can be done in few seconds.

Table 17 shows similarity values are about 1% which means shares and real audio are same. Details of Table 18 show the values are approximately 0 which means that shares are completely randomized and are noisy.

Table 15 Average processing time for audio with random coefficient

Test file	Length (s)	Share creation (s)	Secret reconstruction (s)
Audio1	4	0.5861	0.2878
Audio2	9	0.80423	0.395
Audio3	17	1.1170	0.4747
Audio4	24	1.4612	1.1183

Table 16 Average processing time of audio with random coefficient

Test file	Length (s)	Share creation (s)	Secret reconstruction (s)
Audio1	4	0.76716	0.4067
Audio2	17	1.25703	0.51126
Audio3	9	0.8396	0.37603
Audio4	24	1.0591	0.56616

Table 17 Similarity score for audio with constant coefficients

Sample audio file	Share1	Share2	Reconstructed audio
audio 1	0.9990	0.9973	1.0000
audio 2	1.0000	1.0000	1.0000
audio 3	1.0000	1.0000	1.0000

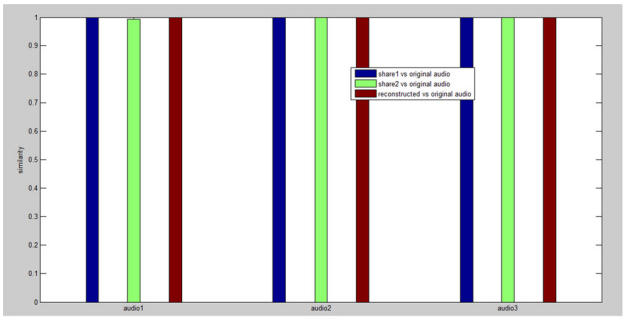
Table 18 Similarity score for audio with random coefficients

Sample audio file	Share1	Share2	Reconstructed audio
audio 1	0.0046	0.0012	1.0000
audio 2	− 0.0039	− 8.7864e−04	1.0000
audio 3	− 0.0054	3.8224e−04	1.0000

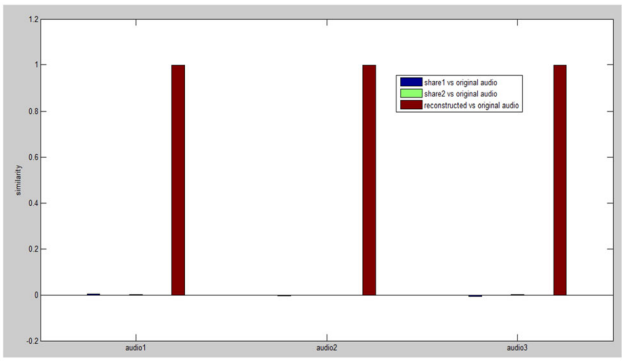
The correlation between original audio, its shares and reconstructed audio is shown in Fig. 16. Results show that in case of constant coefficients, similarity among the original audio and its shares is 1% which means audio signals are highly correlated and hearing of audio shares reveals the information, but when random coefficients are taken, the similarity is nearly 0% which means that the correlation among the samples of audio signal is eliminated and hearing of shares does not reveal any information.

Conclusions

The security of secret in the form of text, image and audio sharing using polynomial-based secret sharing scheme has been analyzed. Shares are generated in two ways first by considering a constant value as the coefficient of the polynomial equation and second by taking random values as the coefficients for each element of secret data. It is found that in all the three forms of the secret, i.e., text, image and audio, the shares generated using constant coefficient are not secured. They reveal the secret as individual shares. But the security of shares is retained when random values are taken as coefficients of the polynomial. The reconstructed secret has negligible loss of information and is very much similar to the original secret, and therefore, there is no need to depend on human visual or auditory system as in visual cryptography.



(a)



(b)

Fig. 16 **a** Similarity plot for audio with constant coefficients; **b** similarity plot for audio with random coefficients

References

1. S. Shivendra, T. Shailendra, K.M. Krishn, Z. Zhigao, K.S. Arun, Providing security and privacy to huge and vulnerable songs repository using visual cryptography. Multimed. Tools Appl. **77**(9), 11101–11120 (2018)
2. S. Washio, Y. Watanabe, Security of audio secret sharing scheme encrypting audio secrets with bounded shares, in *International Conference on Acoustics, Speech and Signal Processing, IEEE, Italy* (2014), pp. 7396–7400
3. A. Shamir, How to share a secret. Commun. ACM **22**(11), 612–613 (1979)
4. M. Abukari Yakubu, C.M. Namunu, K.A. Pradeep, Audio secret management scheme using Shamir's secret sharing, in

International Conference on Multimedia Modeling, Springer link, LNCS 8935 (2015), pp. 396–407
5. C.C. Thien, J.C. Lin, Secret image sharing. Comput. Gr. **26**(5), 765–770 (2002)
6. R. Wang, S. Shyu, Scalable secret image sharing. Sig. Process. Image Commun. **22**(4), 363–373 (2007)
7. D. Yvo, H. Shuang, Q. Jean-Jacques, *Audio and Optical Cryptography*, vol. 1514 (Springer, Berlin, Heidelberg, 1998), pp. 392–404
8. S. Daniel, S.M. Spyros, *General Access Structures in Audio Cryptography*, 6, https://doi.org/10.1109/eit. 2005. 1627018 (2005)
9. L. Huan, Q. Zheng, Z. Xuanping, W. Xu, Auditory cryptography security algorithm with audio shelters. Adv. Control Eng. Inform. Sci. -Proc. Eng. **15**, 2695–2699 (2011)
10. M. Ehdaie, T. Eghlidos, M.R. Aref, A novel secret sharing scheme from audio perspective, in *International Symposium on Telecommunications, IEEE, Tehran* (2008), pp. 13–18