Multimed Tools Appl (2016) 75:8403–8418
DOI 10.1007/s11042-015-2755-6

Audio encryption based on the cosine number transform

Juliano B. Lima[1] **· Eronides F. da Silva Neto**[1]

Received: 2 November 2014 / Revised: 17 April 2015 / Accepted: 15 June 2015 /
Published online: 1 July 2015

Abstract In this paper, we introduce an audio encryption scheme based on the cosine number transform (CNT). The transform, which is defined over a finite field, is recursively applied to blocks of samples of a noncompressed digital audio signal. The blocks are selected using a simple overlapping rule, which provides diffusion of the ciphered data to all processed blocks. A secret-key is used to specify the number of times the transform is applied to each one of such blocks. Computer experiments are carried out and security aspects of the proposed scheme are discussed. Our analysis indicates that the method meets the main security requirements of secret-key cryptography. More specifically, after the encryption of 16-bit audio signals, correlation coefficients significantly close to 0 and entropy values close to 16 were obtained. Furthermore, the flexibility of the method easily allows key space sizes greater than 2^{256} and provides robustness against differential, known-plaintext and chosen-plaintext attacks.

Keywords Multimedia security · Audio encryption · Number-theoretic transform · Cosine number transform

1 Introduction

In the last years, the development of multimedia security techniques has attracted the attention of researchers from several fields of knowledge. This is mainly due to the increasing ease to share digital image, video and audio through communication networks [22]. In this scenario, steganographic and watermarking schemes have the purpose of providing privacy and autenticity using information hiding fundamentals [4, 6, 7,

✉ Juliano B. Lima
juliano_bandeira@ieee.org

1 Department of Electronics and Systems, Federal University of Pernambuco, Av. da Arquitetura, S/N, 50740-550, Recife, Brazil

20, 28]. On the other hand, multimedia encryption schemes employ several strategies and mathematical tools with the purpose of making perceptual and statistical aspects of an image or an audio file seem *noisy* [1, 12, 15, 29, 30]. In the optical domain, image encryption can be performed by using fractional Fourier transforms and random phase encoding, for instance [9, 24]. Chaotic maps, which have the property of being highly sensitive to initial conditions, are also widely used in multimedia encryption techniques [27].

More specifically, digital audio encryption can be implemented in different ways and applied to scenarios with manifold requirements. In [21], for example, chaotic and multiple-keys algorithms are employed in association with discrete transforms; the purpose is to provide a new audio encryption package for TV cloud computing. The audio encryption algorithm proposed in [18] is devoted to online applications and employs transposition and a multiplicative non-binary system. In [8], a higher dimensional chaotic map is used to enhance the key space and the security of a iterative audio encryption method. The operating principle of the approach presented in [17] is on the basis of a virtual optics scheme; both virtual wavelength and virtual diffraction distance are applied in conjunction with a complex-valued random mask to design multiple-locks and multiple-keys in the course of audio data encryption and decryption. In [26], an index-based selective audio encryption for wireless multimedia sensor networks is proposed and, in [11], the authors introduce an advanced partial encryption using watermarking and scrambling in MP3.

In this paper, we propose an audio encryption scheme based on the cosine number transform (CNT). This transform was originally named *finite field cosine transform*, as a reference to the algebraic structures where it is defined [14]. However, in the present context, the CNT can be viewed as a cosine-based version of the number-theoretic transform (NTT), which explains the nomenclature we have adopted. Number-theoretic transforms are well-known in the signal processing community, where they are used in fast algorithms for computing linear convolutions [3, 19] and, more recently, in fragile watermarking techniques [5, 25]. Analogously to the NTT, all arithmetic operations involved in the computation of the CNT are carried out modulo an odd prime p. In other words, computations in extension fields are avoided, which is suitable for signal processing applications.

The first encryption scheme based on the CNT was proposed in [13]. In that paper, the encryption of grayscale images is considered and the secret-key, which is given by a permutation, determines the position of each image block to be processed by the transform. Such a scheme also includes a preliminary transform step, which is not key-dependent. The encryption scheme proposed in the present work is applicable to noncompressed audio. The technique consists in applying the CNT to blocks of samples of an audio signal. The number of times that the transform is recursively applied to each block depends on a secret-key. The transformed block replaces the original block before the next block is processed. Since there is an overlapping among the samples of two adjacent blocks, the ciphered data is diffused along the whole audio signal. This is important to ensure some properties related to the security of the method.

Besides complying with the main security requisites of a multimedia encryption scheme, the proposed approach has the following attractive features: (i) simplicity: basically, the scheme consists in computing transforms of audio blocks and requires only two encryption rounds; (ii) flexibility: the scheme can be easily adapted to audio signals encoded with different numbers of bits per sample and allows adjustments on the key sizes; (iii) fidelity: since rounding is not necessary at any step of the algorithm, if the key is correct, the decrypted audio is identical to the original audio signal; (iv) computational efficiency: the CNT can be computed via fast algorithms and employing fixed-point arithmetic operations only. This permits efficient implementations and reduces the number of additions and multiplications necessary to calculate an N-length CNT from $\mathcal{O}(N^2)$ to $\mathcal{O}(N \log N)$ [3, 10].

This paper is divided as follows. In Section 2, the main theoretical aspects concerning the cosine number transform are presented. In Section 3, we introduce the proposed scheme and describe the steps involved in the encryption/decryption of an audio signal. In Section 4, we present numerical results of computer experiments of the proposed technique and analyze its security. A preliminary comparison between our approach and other state-of-art audio encryption schemes is carried out and some concluding remarks are presented in Section 5.

2 Cosine number transform

The definition of the cosine number transform requires the following finite field cosine function.

Definition 1 Let ζ be a nonzero element in the finite field $\mathrm{GF}(p)$, p an odd prime. The finite field cosine function related to ζ is computed modulo p by

$$\cos_\zeta(x) := \frac{\zeta^x + \zeta^{-x}}{2}, \tag{1}$$

$x = 0, 1, \ldots, \mathrm{ord}(\zeta)$, where $\mathrm{ord}(\zeta)$ denotes de multiplicative order[1] of ζ.

The finite field cosine function holds properties similar to those of the standard real-valued one, such as the unit circle and the addition of arcs, for instance. Definition 1 can also contain additional details, which do not need to be considered in the present context [14]. The cosine number transform is given by the following definition.

Definition 2 Let $\zeta \in \mathrm{GF}(p)$ be an element such that $\mathrm{ord}(\zeta) = 2N$. The cosine number transform of the vector $\mathbf{x} = [x_0, x_1, \ldots, x_{N-1}]$, $x_i \in \mathrm{GF}(p)$, is the vector $\mathbf{X} = [X_0, X_1, \ldots, X_{N-1}]$, $X_j \in \mathrm{GF}(p)$, of elements

$$X_j := \sqrt{\frac{2}{N}} \sum_{i=0}^{N-1} \beta_j x_i \cos_\zeta\left(j \frac{2i+1}{2}\right) \tag{2}$$

computed modulo p, where

$$\beta_j = \begin{cases} 1/\sqrt{2} \pmod{p}, & j = 0, \\ 1, & j = 1, 2, \ldots, N-1. \end{cases}$$

The computation of the CNT of a row vector $\mathbf{x}$ can be represented by the matrix equation

$$\mathbf{X} = \mathbf{C} \cdot \mathbf{x}^T,$$

[1]The multiplicative order of an element ζ in the finite field $\mathrm{GF}(p)$ is the least positive integer l such that $\zeta^l \equiv 1 \pmod{p}$.

where $\mathbf{x}^T$ is the vector $\mathbf{x}$ transpose and $\mathbf{C}$ corresponds to the transform matrix, whose element in the $(j+1)$-th row and the $(i+1)$-th column is given by

$$C_{j+1,i+1} = \sqrt{\frac{2}{N}}\beta_j \cos_\zeta\left(j\frac{2i+1}{2}\right) = \sqrt{\frac{2}{N}}\beta_j \cos_{\sqrt{\zeta}}\left(j(2i+1)\right),$$

$i, j = 0, 1, \ldots, N-1$. It can be shown that the inverse CNT is obtained by using the transform matrix $\mathbf{C}^{-1} = \mathbf{C}^T$ [14]. This means that algorithms and architectures designed to compute a CNT can be easily adjusted to compute the corresponding inverse CNT.

As an example, let us construct a CNT of length $N = 8$ over GF(65537). We use the element $\zeta = 4$, with multiplicative order $\mathrm{ord}(\zeta) = 16$, and, from Definitions 1 and 2, we compute

$$\mathbf{C} = \begin{bmatrix} 1020 & 1020 & 1020 & 1020 & 1020 & 1020 & 1020 & 1020 \\ 24577 & 63491 & 65033 & 65441 & 96 & 504 & 2046 & 40960 \\ 61442 & 65297 & 240 & 4095 & 4095 & 240 & 65297 & 61442 \\ 63491 & 96 & 40960 & 504 & 65033 & 24577 & 65441 & 2046 \\ 64517 & 1020 & 1020 & 64517 & 64517 & 1020 & 1020 & 64517 \\ 65033 & 40960 & 65441 & 63491 & 2046 & 96 & 24577 & 504 \\ 65297 & 4095 & 61442 & 240 & 240 & 61442 & 4095 & 65297 \\ 65441 & 504 & 63491 & 40960 & 24577 & 2046 & 65033 & 96 \end{bmatrix}. \tag{3}$$

Regarding the matrix $\mathbf{C}$ given in (3), it is important to remark that the least positive integer l such that $\mathbf{C}^l = \mathbf{I}$ (the identity matrix) is greater than 10^9. This means that the constructed CNT can be iteratively applied to a vector $\mathbf{x}$ at least 1 billion times before the original vector $\mathbf{x}$ is recovered. Due to this property, the CNT is suitable for cryptographic schemes which use recursive transformations. This strategy would not be effective, for example, if standard number-theoretic transforms were considered, because the fourth power of a standard NTT matrix is equal to the identity matrix [2, 3, 25].

Additionally, it is important to remark that $p = 65537$, the characteristic of the finite field employed in the example developed above, is a Fermat prime, i. e., it has the form $p = 2^{2^s} + 1$, with $s = 4$. In the cases where p is a Fermat prime, the possible multiplicative orders of the elements of GF(p) are divisors of $p - 1 = 2^{2^s}$ and CNT whose lengths are also a power of two can be defined (see Definition 2). This allows to use standard radix-2 decimation-in-time and decimation-in-frequency fast algorithms to compute the CNT [10]. On the other hand, if $p = 2^s - 1$, it is a Mersenne prime. In the cases where p is a Mersenne prime, multiplications by powers of 2 (mod p) correspond to a bit shift. This means that, if the CNT kernel is expressed as a sum of powers of two, multiplication-free transforms can be constructed [3, 16].

Usually, the parameters of a CNT to be used in the processing of a specific signal are chosen in a way such that the computational advantages mentioned above can be achieved. If a signal has samples whose values are integers in the range $0 - M$, for instance, the smallest Fermat (or Mersenne) prime greater than M is selected and a transform defined over the corresponding finite field is constructed. This premise is considered in the design of the encryption scheme introduced in the next section.

3 The encryption scheme

The proposed encryption scheme is illustrated in Fig. 1. It requires the definition of a CNT over the smallest prime finite field in which the range of integer values assumed by the audio

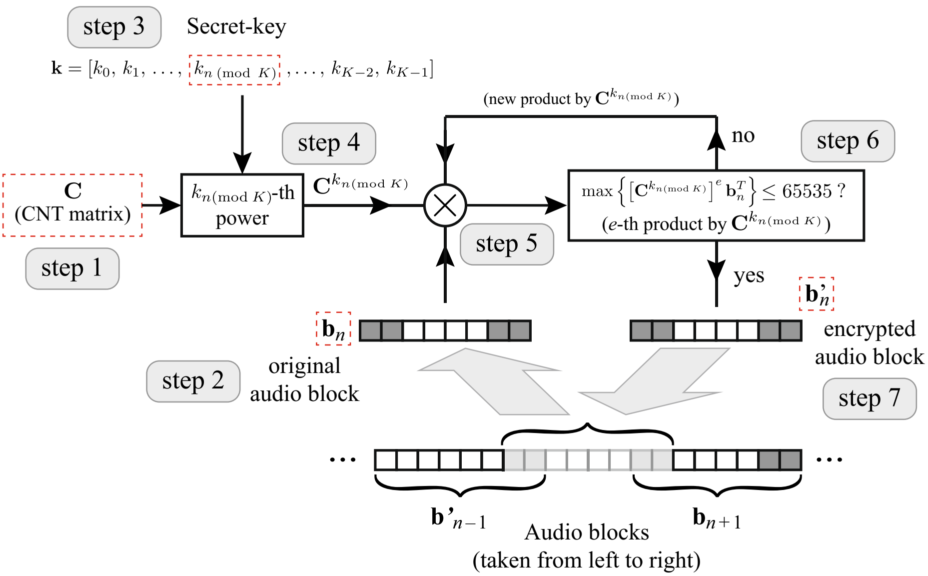

Fig. 1 Block diagram of the proposed audio encryption scheme

samples can be mapped. In this paper, we have designed the proposed encryption scheme to be applied to 16 bits/sample noncompressed audio signals; each sample assumes an integer value in the range $0 - 65535$ and the CNT given in the example presented in Section 2 is used. In step 1 of our scheme, the CNT matrix $\mathbf{C}$ is constructed (see Fig. 1).

An audio block with 8 samples is then taken from the original audio (step 2 in Fig. 1); such a block, which is denoted by $\mathbf{b}_n$ (starting from $\mathbf{b}_1$, which is composed by the first eight samples of the audio signal), is taken in a way such that it overlaps the previous ciphered audio block in two samples; that is, the first two samples of the original audio block $\mathbf{b}_n$ are the last two samples of the ciphered audio block $\mathbf{b}'_{n-1}$ (only the block $\mathbf{b}_1$ is taken without the overlapping). This provides diffusion in our scheme. The index n of the audio block being processed determines the choice of the element of the secret-key used in the scheme (step 3). More specifically, the secret-key is the K-length vector of integers

$$\mathbf{k} = [k_0,\ k_1,\ \ldots,\ k_{K-1}],$$

whose $n \pmod K$-th component is considered in the encryption of the audio block $\mathbf{b}_n$.[2] Such a component determines the computation of the $k_{n \pmod K}$-th power of the CNT matrix $\mathbf{C}$ (step 4). The matrix $\mathbf{C}^{k_{n \pmod K}}$ is then multiplied by the block $\mathbf{b}_n$ (step 5), which produces the provisory ciphered audio block

$$\mathbf{b}'_{n,1} = \mathbf{C}^{k_{n \pmod K}} \cdot \mathbf{b}_n^T. \tag{4}$$

This corresponds to compute $k_{n \pmod K}$ times the CNT of $\mathbf{b}_n$ in an iterative manner.

[2]The index of the component selected in the secret-key has to be reduced modulo K because the number of audio blocks to be processed throughout the encryption procedure is usually greater than the key-length K. In this sense, the K-th block is processed using the component of index $K \pmod K \equiv 0 \pmod K$ of the secret-key; the $(K+1)$-th block is processed using the component of index $K + 1 \pmod K \equiv 1 \pmod K$ of the secret-key and so on.

The block computed in (4) is provisory because, since all computations are carried out modulo $p = 65537$, $\mathbf{b}'_{n,1}$ may contain samples whose values are equal to 65536. This would require a binary representation with 17 bits, which violates the encoding of the original audio signal. In order to avoid such an extra bit, the matrix $\mathbf{C}^{k_n \pmod K}$ is iteratively multiplied by $\mathbf{b}_n$; if a block $\mathbf{b}'_{n,e}$ obtained from (4) in the e-th iteration contains a sample equal to 65537, we update such a block, multiplying it by $\mathbf{C}^{k_n \pmod K}$ again. The process stops when a *new* block $\mathbf{b}'_{n,E}$ without samples equal to 65536 is encountered in the E-th iteration (step 6). The definitive block $\mathbf{b}'_n = \mathbf{b}'_{n,E}$ is then taken as the encrypted version of $\mathbf{b}_n$ and replaces $\mathbf{b}_n$ in the composition of the encrypted audio signal (step 7). The encryption procedure is completed after the whole audio vector is submitted to two rounds of the described transformation strategy. This is necessary to make brute-force attacks unfeasible.

The decryption consists in applying, in the reverse order, the same steps used in the encryption; the matrix $\mathbf{C}$ is replaced by the the matrix $\mathbf{C}^{-1} = \mathbf{C}^T$ and the blocks are taken from right to left. We remark that the number of times each audio block has to be iteratively multiplied by $\mathbf{C}^{k_n \pmod K}$ in the encryption does not need to be known for a successful decryption. Suppose that

$$\left(\mathbf{b}'_{n,e}\right)^T = \left[\mathbf{C}^{k_n \pmod K}\right]^e \cdot \mathbf{b}_n^T, \quad e = 1, 2, \ldots, E-1,$$

contains at least one sample whose value is equal to 65536, but the maximum sample value in

$$\left(\mathbf{b}'_{n,E}\right)^T = \left[\mathbf{C}^{k_n \pmod K}\right]^E \cdot \mathbf{b}_n^T$$

does not exceed 65535. Then, $\mathbf{b}'_{n,E} = \mathbf{b}'_n$ is taken as the (definitive) ciphered version of $\mathbf{b}_n$. In the decryption,

$$\left(\mathbf{b}'_{n,E-d}\right)^T = \left[\mathbf{C}^{-k_n \pmod K}\right]^d \cdot \left(\mathbf{b}'_n\right)^T = \left[\mathbf{C}^{-k_n \pmod K}\right]^d \cdot \left[\mathbf{C}^{k_n \pmod K}\right]^E \cdot \mathbf{b}_n^T,$$

$d = 1, 2, \ldots, E-1$, will contain at least one sample whose value equal to 65536; actually, $(\mathbf{b}'_{n,E-d})^T$, $d = 1, 2, \ldots, E-1$, are the same provisory blocks produced in the encryption. Only when $d = E$ is reached, a block which does not contain at least one sample value equals 65536 is obtained. Such a block is

$$\left(\mathbf{b}'_{n,0}\right)^T = \left[\mathbf{C}^{-k_n \pmod K}\right]^E \left[\mathbf{C}^{k_n \pmod K}\right]^E \cdot \mathbf{b}_n^T = \mathbf{b}_n^T,$$

the original audio block correctly recovered.

4 Computer experiments and security analysis

The proposed encryption scheme was implemented in Matlab®. Segments with 1.8×10^5 samples of eight noncompressed audio signals with different characteristics (music, speech etc.), encoded with 16 bits/sample, were encrypted using

$$\mathbf{k} = [25,\ 34,\ 224,\ 146,\ 16,\ 60,\ 91,\ 210,\ 4,\ 11,\ 44,\ 166,\ 187,\ 166,\ 115] \tag{5}$$

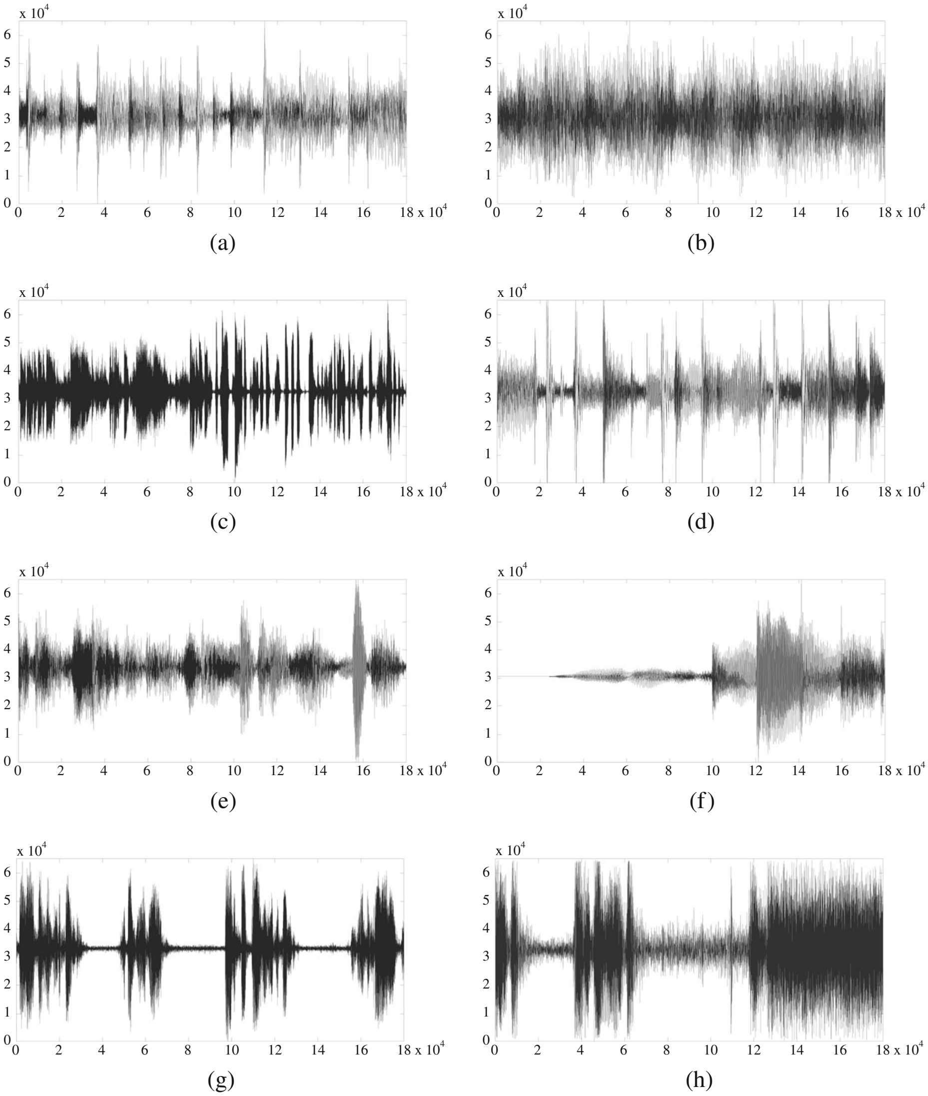

Fig. 2 Original audio signals used in the computer experiments: (**a**) audio_01.wav, (**b**) audio_02.wav, (**c**) audio_03.wav, (**d**) audio_04.wav, (**e**) audio_05.wav, (**f**) audio_06.wav, (**g**) audio_07.wav, (**h**) audio_08.wav

as secret-key. In Fig. 2, the waveforms of the original audio signals are shown. The file audio_01.wav, for example, was obtained using the sampling rate $F_s = 8000$ Hz and, therefore, has time duration equal to 22.5 s; nevertheless, the application of the encryption procedure independs on the sampling rate used in the discretization of a signal.

In Fig. 3a, the complete ciphered version of audio_01.wav is shown. Naturally, the "dense" visual aspect of the waveform reflects the rapid variations arising from the encryption process. In order to provide a more suitable visualization, we show in Fig. 3b the first

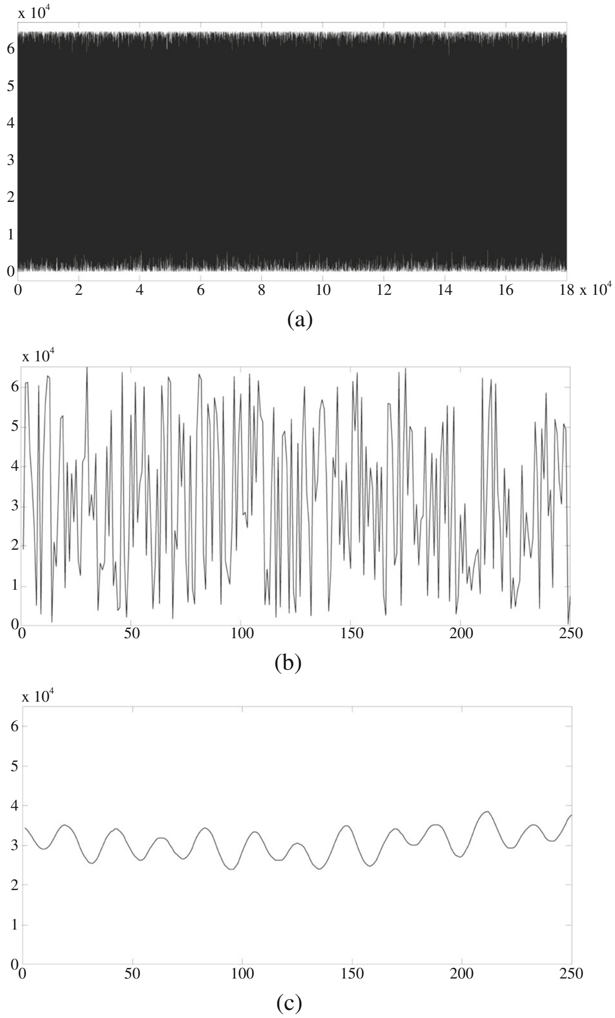

Fig. 3 (**a**) Full ciphered version of audio_01.wav and (**b**) its first 250 samples; (**c**) first 250 samples of the original version of audio_01.wav

250 samples of the same ciphered audio signal (the samples are connected with lines to enhance the visualization). In this figure, we can observe the noisy aspect of the waveform. This contrasts with the waveform presented in Fig. 3c, where the first 250 samples of the corresponding original audio signal are shown. In this case, the quasi-periodicity of the waveform is emphasized; this property indicates that the non-ciphered audio segment may represent, for instance, a voiced speech. The visual aspect of the waveforms corresponding to ciphered versions of all other audio signals used in our experiments is similar.

4.1 Histogram analysis

The noisy aspect of the ciphered versions of the audio signals is also reflected in their histograms. In Fig. 4a, the histogram of audio_03.wav is shown; it follows a specificdistribution, which is similar to the distributions observed for the other original audio signals. On the other hand, the histogram of the ciphered version of audio_03.wav (Fig. 4b) has a flat shape. This behavior is also verified for the other audio signals. Since the number of samples in the audio segments employed in the simulations (1.8×10^5) is relatively small when compared with the number of symbols in the *source alphabet* (65536), the histogram in Fig. 4b appears not to be uniform. However, if longer audio segments are used, the tendency

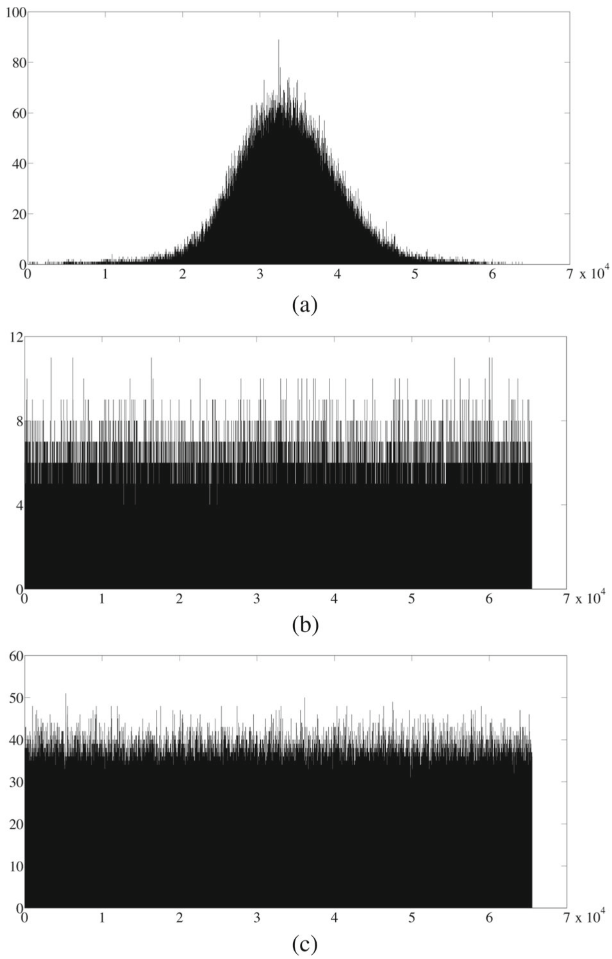

Fig. 4 Histograms of the original version (**a**) and the ciphered version (**b**) of a segment with 1.8×10^5 samples of audio_03.wav; (**c**) histogram of the ciphered version of a segment with 1.8×10^6 samples of audio_03.wav. The histograms shown in (**b**) and (**c**) has a uniform tendency

of uniformization can be observed. See, for example, Fig. 4c, where the histogram of the ciphered version of a segment with 1.8×10^6 samples of audio_03.wav is shown. This suggests that the proposed encryption scheme produces samples uniformly distributed and weakly correlated.

4.2 Statistical analysis

An objective analysis of the statistical properties of the ciphered audio signals resulting from our experiments can be carried out by computing correlation coefficients. By selecting arbitrarily P samples, the correlation coefficient is computed by

$$r_{xy} = \frac{\mathrm{cov}(x, y)}{\sqrt{D(x)D(y)}},$$

where $\mathrm{cov}(x, y) = \frac{1}{P}\sum_{i=1}^{P}(x_i - E(x))(y_i - E(y))$, $D(x) = \frac{1}{P}\sum_{i=1}^{P}(x_i - E(x))^2$ and $E(x) = \frac{1}{P}\sum_{i=1}^{P}x_i$; x_i is the value of the i-th selected sample and y_i is the value of the corresponding adjacent sample. The results for $P = 10^5$ are shown in Table 1. Original audio signals have correlation coefficients clearly close to one, while ciphered audio signals have correlation coefficients close to zero. This indicates that the proposed scheme is resistant against statistical attacks. Moreover, in the simulations, the entropy of the ciphered audio files has assumed values varying from 15.7057 to 15.7117. Although these values are greater than those commonly observed for non-ciphered 16-bit audio data, they are not too close to 16. Again, this is due to the relationship between the number of samples of the audio signals used in our experiments and the number of symbols in the *source alphabet*, that is, 65536. If we consider a segment with 1.8×10^6 samples of audio_03.wav (10 times longer than the segment employed in our simulations), for instance, the corresponding ciphered audio has entropy equal to 15.9735, which is significantly closer to 16, when compared with the entropy values previously given. A similar behavior is verified for all other audio signals. This means that the transformed audio signals are close to a random source and the proposed technique is also secure against the entropy attack.

4.3 Key space

Other important security parameter is the key space. If we encode each key position with 10 bits, for example, a key space of size 2^{150} is achieved with the 15-length key used in the experiments ($K = 15$). Under this aspect, the proposed scheme is very flexible. Larger key

Table 1 Correlation coefficients of original (r) and ciphered ($\tilde{r}$) audio files used in the simulations

File name	r	$\tilde{r}$
audio_01.wav	0.6405	0.0021
audio_02.wav	0.9982	0.0036
audio_03.wav	0.9804	−0.0018
audio_04.wav	0.9941	−0.0042
audio_05.wav	0.9934	−0.0026
audio_06.wav	0.9989	0.0087
audio_07.wav	0.9677	0.0045
audio_08.wav	0.9830	0.0057

spaces can be obtained increasing the key length or increasing the number of bits used to encode each key position. A key space of size 2^{256}, for example, is obtained if we consider $K = 16$ and encode each key position with 16 bits. In fact, according to the comments made after (3), each key position could be an integer in the range $1 - 10^9$. This indicates that our scheme is secure against brute-force attacks [23].

4.4 Robustness to differential attacks

In order to evaluate the resistance of the method against differential attacks, for each original audio signal, we randomly choose one sample. The least significant bit of such a sample is inverted and a modified audio signal is obtained. Original and modified audio signals are encrypted using the same key and two ciphered audio signals are generated. Such ciphered audio signals are then compared by the number of samples change rate (NSCR) and the unified average changing intensity (UACI), which are defined by [27]

$$\mathrm{NSCR} = \frac{\sum_i D_i}{L} \times 100\ \%$$

and

$$\mathrm{UACI} = \frac{1}{L}\left[\sum\nolimits_i \frac{|A_i - A_i'|}{65535}\right].$$

A and A' are the two ciphered audio signals whose corresponding original audio signals have only one-sample difference; the values of the samples at position i of A and A' are respectively denoted by A_i and A_i'; L corresponds to the length of the audio vector; D_i is determined according to the rule

$$D_i = \begin{cases} 1, & A_i \neq A_i', \\ 0, & \text{otherwise.} \end{cases}$$

The ideal values for NSCR and UACI are 100 % and 33.3 %, respectively [27]. In Table 2, the minimum, the maximum and the average values of NSCR and UACI, computed from the encryption of 100 different modified versions of each audio signal are shown. The results are considerably close to the ideal values and independ on the position of the modified sample.

4.5 Key sensitivity

The key sensitivity of the proposed encryption scheme is evaluated by encrypting an audio signal with a given secret-key $\mathbf{k}$ and attempting to decrypt it with a *wrong* key $\mathbf{k}'$ slightly different from $\mathbf{k}$. In our simulations, we use

$$\mathbf{k}' = [25,\ 34,\ 224,\ 146,\ 16,\ 60,\ 91,\ 210,\ 4,\ 11,\ 44,\ 166,\ 187,\ 166,\ \underline{114}]$$

which is different from the key $\mathbf{k}$ given in (5) by one bit only (the underlined number 114 in $\mathbf{k}'$ replaces the number 115 in $\mathbf{k}$). The original audio and that recovered with $\mathbf{k}'$ are compared using the number of samples change rate. The NSCR obtained for all audio signals used in the simulations vary from 99.9972 % to 100.0000 %, which means that the audio signals decrypted using the *wrong* key are completely different from the original ones. In fact, the aspect of the recovered audio signals is completely noisy, being similar to those presented in Figs. 3a and 3b.

Table 2 Maximum, minimum and average NSCR and UACI (100 different modified versions of each audio signal were used)

File name	Metric	Max (%)	Min (%)	Avg. (%)
audio_01.wav	NSCR	100.0000	99.9967	99.9992
	UACI	33.4091	33.1850	33.2924
audio_02.wav	NSCR	100.0000	99.9973	99.9992
	UACI	33.4743	33.1931	33.3012
audio_03.wav	NSCR	100.0000	99.9961	99.9992
	UACI	33.4770	33.1858	33.3570
audio_04.wav	NSCR	100.0000	99.9967	99.9993
	UACI	33.4367	33.2030	33.3196
audio_05.wav	NSCR	100.0000	99.9961	99.9993
	UACI	33.4562	33.1830	33.3108
audio_06.wav	NSCR	100.0000	99.9972	99.9992
	UACI	33.3864	33.1391	33.2898
audio_07.wav	NSCR	100.0000	99.9972	99.9992
	UACI	33.4137	33.1590	33.3122
audio_08.wav	NSCR	100.0000	99.9972	99.9991
	UACI	33.4941	33.2368	33.3706

4.6 Known-plaintext and chosen-plaintext attacks

A preliminary analysis indicates that the proposed scheme can also resist to known-plaintext and chosen-plaintext attacks. Even if an adversary has access to some plaintext/ciphertext pair, the overlapping among adjacent audio blocks and the employment of a two-round encryption procedure reduce to a brute-force attack the attempt of obtaining the secret-key. The adversary could find the position ℓ of the last ciphered audio block $\mathbf{b}'_\ell$ and also determine the index $\ell \pmod K$ of the component $k_{\ell \pmod K}$ of the secret-key $\mathbf{k}$ used to encrypt such a block. However, $k_{\ell \pmod K}$ could not be discovered by "comparing" successive results of recursive computations of the inverse CNT of $\mathbf{b}'_\ell$ to the corresponding known-plaintext (the block $\mathbf{b}_\ell$ at the same position in the original audio). This is due to the fact that, even if the decryption of $\mathbf{b}'_\ell$ is correct, it produces an audio block which is composed by blocks encrypted in the first encryption round.

The choice of plaintexts that would reveal the secret-key is apparently not straight. If we choose an audio signal with all the samples equal to zero, a ciphered audio signal with all the samples equal to zero is obtained. Another possibility is to choose an audio such that the only nonzero sample is the first one, i. e., $\mathbf{b}_1 = [1\ 0\ 0\ 0\ 0\ 0\ 0\ 0]$ and $\mathbf{b}_i = \mathbf{0}$, $i \neq 1$. If an adversary has access to $\mathbf{b}'_1$ (computed in the first encryption round), he can obtain the exponent k_1 by verifying exhaustively whether the first column of $\mathbf{C}^{k_1}$ is equal to $\left(\mathbf{b}'_1\right)^T$. Applying a similar procedure to the pair $\mathbf{b}_2$ and $\mathbf{b}'_2$, k_2 could be obtained and so on. However, such a procedure is feasible only if the adversary has access to each ciphered audio block before the next block is processed; usually, this is not considered a realistic attack scenario. If the adversary has access only to the whole ciphered audio, even if he can choose a plaintext, the described attack becomes impractical.

5 Discussion and concluding remarks

We have introduced an audio encryption scheme based on the cosine number transform. The scheme is very flexible and can be applied to noncompressed digital data encoded with different numbers of bits per sample. Our approach has demonstrated robustness against the main cryptographic attacks, namely, statistical, brute-force, differential, known-plaintext and chosen-plaintext attacks [23]. This is ratified by the results obtained in our experiments, which include the calculation of several metrics specifically related to certain types of attacks. In the literature, the contexts in which other audio encryption techniques are placed are very diverse. Furthermore, the incompleteness of some previous audio encryption papers with respect to security aspects makes unfeasible a systematic and full comparison with our approach. Considering such restrictions, we carry out a preliminary and qualitative analysis regarding this point.

In [18], for example, the authors basically compare the histogram of one audio signal to the histogram of the corresponding ciphered audio signal by means of a visual inspection. Moreover, the entropy, the standard deviation and the mean absolute difference of such a ciphered audio are calculated. Although such measurements indicate that the ciphered audio has a statistical behavior similar to that of a uniformly distributed random source, they are not sufficient to ensure the security of the method.

The scheme proposed in [17] is based on virtual optics and requires the conversion of the audio signal into a two-dimensional sound map. According to the authors, the method is highly sensitive to deviations in parameters related to the secret-key and its key space size can reach $(2^8)^{256\times256}$. Besides not presenting a complete security analysis, the implementation of the method depends on the knowledge of operations and elements commonly employed in optics frameworks, but probably unfamiliar in multimedia scenarios. This hinders its practical utilization, which contrasts with the straightness and the suitability of our method for processing digital data.

The method described in [21] employs real-valued discrete transforms, which means that rounding operations are necessary. Unlike our method, this may produce a decrypted audio signal slightly different from the corresponding original audio signal. Moreover, the experiments performed by the authors are incomplete (for example, only one audio signal is considered) and nonusual metrics are computed for security analysis. Nevertheless, we have verified that our correlation measurements are similar to those obtained by the authors in their paper.

In [8], a scheme based on a higher dimensional Arnold's cat map is proposed. According to the authors, the security level of their approach depends on the iteration time, which is directly proportional to the key space size. On the other hand, our scheme involves only two rounds and the key space size can be increased without the need of more iterations. Moreover, although the authors mention several security parameters, they do not present numerical results. This raises doubts regarding the robustness of the method against specific cryptographic attacks.

The scopes considered in [11] and [26] are quite different from ours. In [11], a partial encryption is perfomed by means of watermarking and scrambling in MP3; the authors perform numerical experiments whose focus is the overhead rate and the watermark robustness against amplitude reduction and echo addition. The selective encryption scheme presented in [26] encrypts only the important audio data in order to achieve both real-time performance and energy efficient transmission in wireless multimedia sensor networks; security aspects and metrics usually considered to evaluate a data encryption algorithm are not discussed.

The extension of the proposed scheme to digital images is currently under investigation. In this case, a two-dimensional CNT has to be considered and the transform parameters have to be chosen according to specific digital image standards. Aspects related to the fast computation of the CNT should also be considered in future work. This is particularly important in scenarios where hardware implementations of the proposed encryption technique are desired. The definition of a CNT over fields of characteristic two is also part of the topics for future research. The possibility of defining a CNT over GF(2^{16}), for instance, would eliminate the need of recursively computing the transform of an audio block in order to avoid the appearance of certain sample values. This would simplify and make our method faster.

Acknowledgments This research was supported by Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) under Grants 307686/2014-0 and 456744/2014-2.

References

1. Abuturab MR (2013) Color image security system based on discrete hartley transform in gyrator transform domain. Opt Lasers Eng 51(3):317–324
2. Birtwistle DT (1982) The eigenstructure of the number theoretic transforms. Signal Process 4(4):287–294
3. Blahut RE (2010) Fast algorithms for signal processing. Cambridge University Press
4. Cheddad A, Condell J, Curran K, McKevitt P (2010) Digital image steganography: Survey and analysis of current methods. Signal Process 90(3):727–752
5. Cintra RJ, Dimitrov VS, Campello de Souza RM, de Oliveira HM (2009) Fragile watermarking using finite field trigonometrical transforms. Signal Process Image Commun 24:587–597
6. Cox I, Miller M, Bloom J, Fridrich J, Kalker T (2007) Digital watermarking and steganography, 2nd edn. The Morgan Kaufmann series in multimedia information and systems. Morgan Kaufmann
7. Fallahpour M, Megias D (2009) High capacity audio watermarking using FFT amplitude interpolation. IEICE Electron Express 6(14):1057–1063
8. Gnanajeyaraman R, Prasadh K, Ramar D (2009) Audio encryption using higher dimensional chaotic map. Int J Recent Trends Eng 1(2):103–107
9. Gong L, Liu X, Zheng F, Zhou N (2013) Flexible multiple-image encryption algorithm based on log-polar transform and double random phase encoding technique. J Modern Opt 60(13):1074–1082
10. Kok CW (1997) Fast algorithm for computing discrete cosine transform. IEEE Trans Signal Process 45(3):757–760
11. Kwon GR, Wang C, Lian S, Hwang SS (2012) Advanced partial encryption using watermarking and scrambling in MP3. Multimed Tools Appl 59(3):885–895
12. Lian S (2008) Multimedia content encryption: techniques and applications, 7th edn. Auerbach Publications
13. Lima JB, Lima EAO, Madeiro F (2013) Image encryption based on the finite field cosine transform. Signal Process Image Commun 28(10):1537–1547
14. Lima JB, Campello de Souza RM (2011) Finite field trigonometric transforms. Appl Algebra Eng Commun Comput 22(5-6):393–411

15. Madain A, Abu Dalhoum AL, Hiary H, Ortega A, Alfonseca M (2014) Audio scrambling technique based on cellular automata. Multimed Tools Appl 71(3):1803–1822
16. Nibouche O, Boussakta S, Darnell M (2009) Pipeline architectures for radix-2 new Mersenne number transform. IEEE Trans Circ Syst–I: Regular Papers 56(8):1668–1680
17. Peng X, Cui Z, Cai L, Yu L (2003) Digital audio signal encryption with a virtual optics scheme. Optik - Int J Light Electron Opt 114(2):69–75
18. Raghunandhan kR, Radhakrishna D, Sudeepa KB, Ganesh A (2013) Efficient audio encryption algorithm for online applications using transposition and multiplicative non-binary system. Int J Eng Res Technol 2(6):472–477
19. Rubanov NS, Bovbel EI, Kukharchik PD, Bodrov VJ (1998) The modified number theoretic transform over the direct sum of finite fields to compute the linear convolution. IEEE Trans Signal Process 46(3):813–817
20. Sadek MM, Khalifa AS, Mostafa MGM (2014) Video steganography: a comprehensive review. Multimed Tools Appl. 1–32. doi:10.1007/s11042-014-1952-z
21. Serag Eldin SM, Khamis SA, Mahmoud Hassanin AAI, Alsharqawy MA (2015) New audio encryption package for TV cloud computing. Int J Speech Technol 18(1):131–142
22. Shih FY (2012) Multimedia security: watermarking, steganography and forensics. CRC Press
23. Smart N (2011) ECRYPT II yearly report on algorithms and keysizes (2010-2011). Tech. rep., European Network of Excellence in Cryptology II
24. Sui L, Duan K, Liang J, Zhang Z, Meng H (2014) Asymmetric multiple-image encryption based on coupled logistic maps in fractional Fourier transform domain. Opt Lasers Eng 62:139–152
25. Tamori H, Yamamoto T (2009) Asymetric fragile watermarking using a number theoretic transform. IEICE Trans Fundament Electron Commun Comput Sci E92-A(3):836–838
26. Wang H, Hempel M, Peng D, Sharif H, Chen HH (2010) Index-based selective audio encryption for wireless multimedia sensor networks. IEEE Trans Multimed 12(3):215–223
27. Wang Y, Wong KW, Liao X, Chen G (2011) A new chaos-based fast image encryption algorithm. Appl Soft Comput 11(1):514–522
28. Yan D, Wang R, Yu X, Zhu J (2012) Steganography for MP3 audio by exploiting the rule of window switching. Comput Secur 31(5):704–716
29. Ye G (2010) Image scrambling encryption algorithm of pixel bit based on chaos map. Pattern Recog Lett 31(5):347–354
30. Zhou N, Zhang A, Zheng F, Gong L (2014) Image compression-encryption hybrid algorithm based on key-controlled measurement matrix in compressive sensing. Opt Laser Technol 62:152–160

Juliano B. Lima was born in Brazil, where he studied Electrical Engineering. He received the M.Sc. degree and the D.Sc. degree from the Federal University of Pernambuco (Brazil). Currently, he is an Assistant Professor at the Department of Electronics and Systems of the Federal University of Pernambuco in Recife (Brazil). His main research interests are in Cryptography, finite fields and applications, discrete transforms and fast algorithms for digital signal processing.

Eronides F. da Silva Neto was born in Brazil. Currently, he is an undergraduate student in Electronics Engineering at the Federal University of Pernambuco. Since June 2014, he has been part of a oneyear internship at the Temple University, Philadelphia, PA. His main research interests are in multimedia security, information hiding and fast algorithms for digital signal processing.