# Performances of Random Codes Based on Quasigroups for transmission of audio files

Aleksandra Popovska-Mitrovikj, Vladimir Ilievski and Verica Bakeva

***Abstract* — Error-correcting codes can be applied for transmission of images and audio files through a noise channel. Application of Random Codes Based on Quasigroups (RCBQ) for image transmission is considered elsewhere. In this paper we investigate performances of these codes for transmission of audio files through a binary-symmetric channel. We present and analyze several experimental results obtained using Cut-Decoding and 4-Sets-Cut-Decoding algorithms defined for RCBQ. In the experiments we use code (72, 576) with rate 1/8. We made experiments with Beethoven`s "Ode to Joy".**

***Keywords* — random code, cryptcoding, error-correcting code, packet-error probability, bit-error probability, audio file.**

## I. Introduction

Crypt-codes are error-correcting codes resistant to an intruder attack. Usually these codes are obtained with application of some of the known ciphers on the codewords, before sending them through an insecure channel ([1,2]). In this case, two algorithms are used, one for error-correcting codes and another for obtaining information security. In order to obtain more efficient design, in the paper [3] authors give one algorithm where a block cipher and an error-correcting code are combined.

For the first time, Random Codes Based on Quasigroups (RCBQ) are proposed in [4]. These random error-correcting codes are defined by using a cryptographic algorithm during the encoding/decoding process. Therefore, they allow not only correction of certain amount of errors in transmitted data, but they also provide an information security, all built in one algorithm.

The RCBQ are designed using algorithms for encryption/decryption from the implementation of TASC (Totally Asynchronous Stream Ciphers) by quasigroup string transformations (see [5]). These cryptographic algorithms use the alphabet $Q$ and a quasigroup operation * on $Q$ together with its parastrophe \. From the definition of the algorithms it is clear that in their design can be used other algorithms for encryption and decryption**.**

In [6] authors have investigated the influence of the code parameters to the code performances. In [7] and [8] Cut-Decoding and 4-Sets-Cut-Decoding algorithms are proposed such that the modified decoding process is approximately 5.2 (i.e., 6.3) times faster than the original one for code (72,576).

The decoding of RCBQ is actually a list decoding. Therefore the speed of the decoding process and the probability of correct decoding depend on the size of the lists with the possible candidates for decoded message. In order to improve the decoding speed, A.Popovska-Mitrovikj et al. have proposed Cut-Decoding algorithm ([7]), where two transformation of the redundant message with different parameters are used, and the candidates for the decoded messages are obtained by intersection of the corresponding decoding candidate sets (or lists). In this way, we have obtained 4.5 times faster decoding process than the original one for code (72,288). This improvement of the decoding speed gave us an idea of using cuts of more sets, in order to obtain greater increase of the decoding speed. In [8], we proposed a few modifications of Cut-Decoding, called 4-Sets-Cut-Decoding algorithms. In these algorithms we use four transformations of the redundant message and intersections of four decoding candidate sets and we obtain greater improvement of the decoding speed. Also, in order to reduce the number of unsuccessful decodings we have defined several modifications (or versions) of this decoding algorithm. With the third version of 4-Sets-Cut-Decoding algorithm, the best results for packet-error probability (*PER*) and bit-error probability (*BER*) are obtained. From the duration of our experiments with code (72, 576), we concluded that for this code, Cut-Decoding algorithm is 5.2 faster than the original algorithm (given in [4]), and 4-Sets-Cut-Decoding algorithms are 6.3 times faster.

Since, the decoding process in all algorithms for RCBQ is actually a list decoding, the decoding can finish early (if the list of candidates is empty) and then we have a *null-error* or it can finish with more than one element in the list, then a *more-candidate-error* is obtained. In order to reduce the number of these types of error, in [6] and [8] we proposed two methods by backtracking. In these methods we cancel some iterations and we reprocess the first of canceled iterations with larger value (for decreasing the number of *null-errors*) or smaller value of predicted bit-errors $B_{max}$ in a block (for decreasing the number of *more-candidate-errors*).

In this paper, we investigate performances of RCBQ for transmission of audio files through a binary-symmetric channel.

This research was partially supported by Faculty of Computer Science and Engineering at the University "Ss Cyril and Methodius" in Skopje

Aleksandra Popovska-Mitrovikj, Vladimir Ilievski, Verica Bakeva are with the Faculty of Computer Science and Engineering, University "Ss Cyril and Methodius" - Skopje, P.O. Box 393, R. of Macedonia (phone: +389-71-277039; e-mails: {aleksandra.popovska.mitrovikj, verica.bakeva}@finki.ukim.mk, ilievski.vladimir@live.com.

## II. Experimental results

In this section we present the experimental results obtained by transmission of audio files. In our experiments we use binary-symmetric channel and RCBQ as an error-correcting code. We compare the results obtained using both algorithms for RCBQ: Cut-Decoding and 4-Sets-Cut-Decoding. In both algorithms we use the proposed methods with backtracking for reducing the number of errors. All experiments are made for (72, 576) code with rate 1/8 with $B_{max} = 5$ and the parameters given in [7].

We use the audio signal that is consisted of one 16-bit channel with a sampling rate of 44100 *Hz* and it is a part of the Beethoven's "Ode to joy" with a total length of approximately 4.3 seconds.

In all experiments (for different values of bit-error probability $p$ in the channel) we consider: the differences between the sample values of the original and transmitted signal and we compute *BER* and *PER*.

The experimental results for a bit-error probabilities $p = 0.05$, $p = 0.08$, $p = 0.11$, $p = 0.14$ and $p = 0.17$, using Cut-Decoding and 4-Sets-Cut-Decoding algorithm (the third version) are presented and compared. In all graphical presentation (Fig.1–8) the difference between the audio signals (original and transmitted through the channel) is presented. There, the number of the sample in the sequence of samples consisting the audio signal is on the $x$-axis and the value of the sample is on the $y$-axis. The original audio samples are colored in red, and the transmitted audio samples are colored in blue.

For $p = 0.05$, the difference between the original and transmitted audio signal obtained using Cut-Decoding algorithm and 4-Sets-Cut-Decoding algorithm are given on Fig. 1 and Fig. 2, correspondingly.

It is evident from the figures, that for this probability, the result for 4-Sets-Cut-Decoding algorithm is better than the result for Cut-Decoding algorithm.

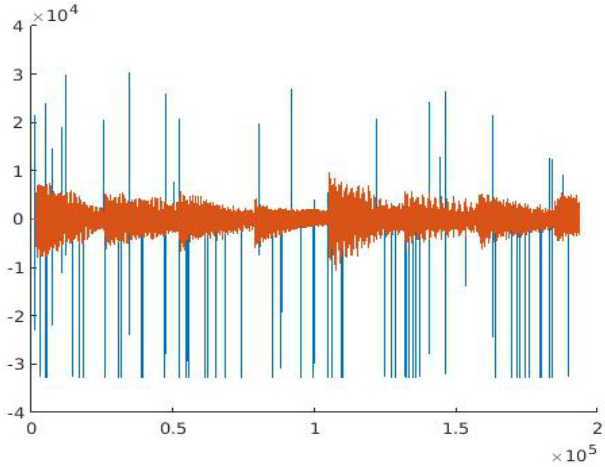


Fig. 1. Difference between the original and transmitted audio signal with Cut-Decoding for $p = 0.05$

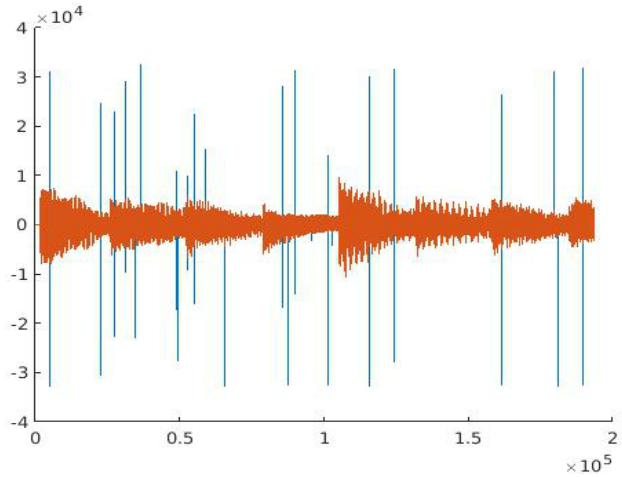


Fig. 2. Difference between the original and transmitted audio signal with 4-Sets-Cut-Decoding for $p = 0.05$

For bit-error probability $p = 0.08$, the results using both algorithms are presented in Fig. 3 and Fig. 4, respectively.

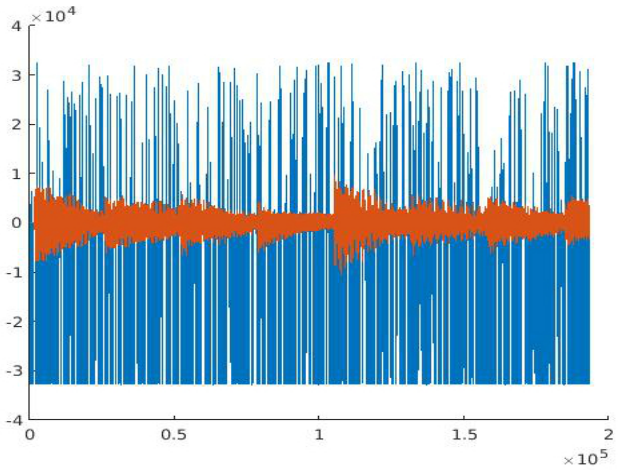


Fig. 3. Difference between the original and transmitted audio signal with Cut-Decoding for $p = 0.08$

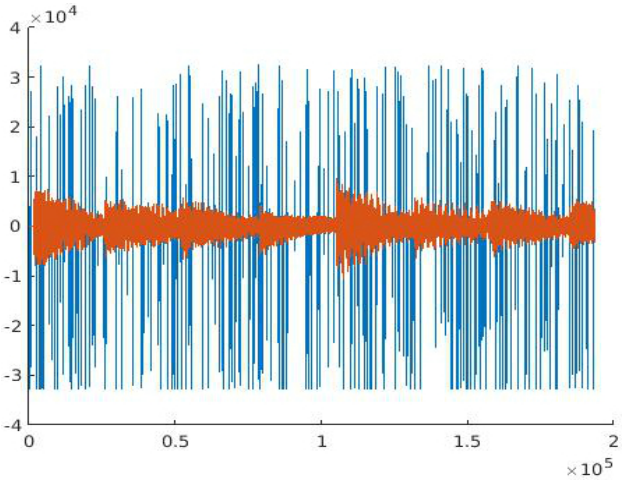


Fig. 4. Difference between the original and transmitted audio signal with 4-Sets-Cut-Decoding for $p = 0.08$

Now, the number of incorrectly decoded messages increases, but from the graphs we can notice that 4-Sets-

Cut-Decoding algorithm gives better result than Cut-Decoding algorithm.

The results for $p = 0.11$, using both algorithms are presented in Fig. 5 and Fig. 6.

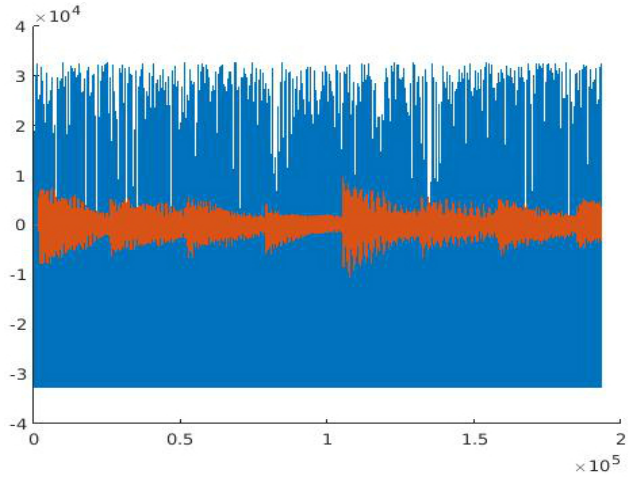


Fig. 5. Difference between the original and transmitted audio signal with Cut-Decoding for $p = 0.11$

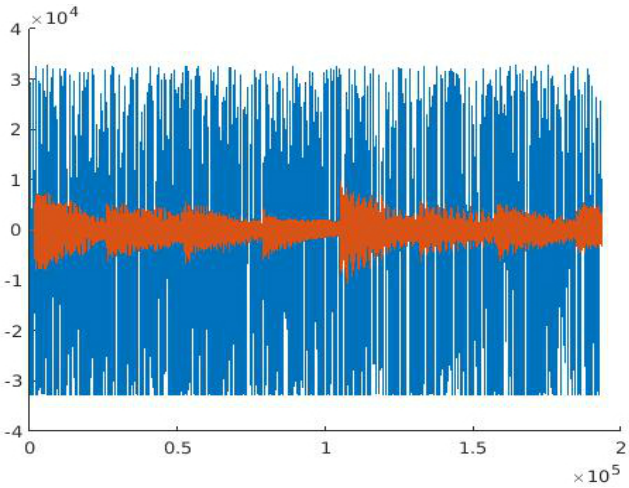


Fig. 6. Difference between the original and transmitted audio signal with 4-Sets-Cut-Decoding for $p = 0.11$

From Fig. 5 we can see that the transmitted signal using Cut-Decoding algorithm is too much noisy, and for each bit-error probability greater than 0.11, we concluded that *BER* is greater than the bit-error probability in the channel, so no sense to make further experiments with this algorithm. For this reason, we do not show the results obtained by Cut-Decoding algorithm for bit-error probabilities 0.14 and 0.17.

For bit-error probability $p = 0.14$ and $p = 0.17$ the difference between the original and transmitted signal using 4-Sets-Cut-Decoding algorithm is presented in Fig. 7 and Fig. 8, correspondingly.

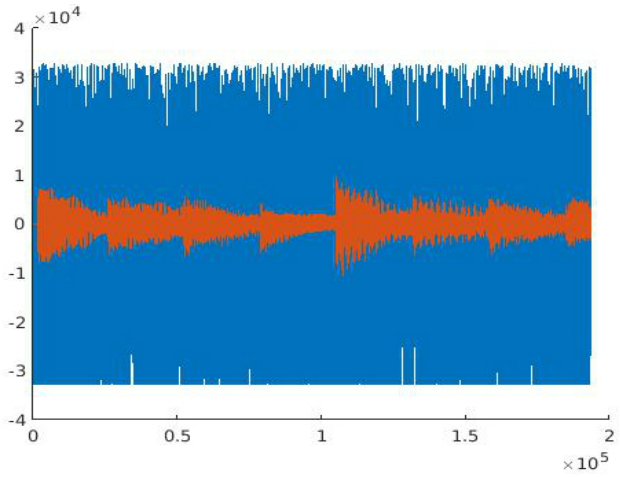


Fig. 7. Difference between the original and transmitted audio signal with 4-Sets-Cut-Decoding for $p = 0.14$

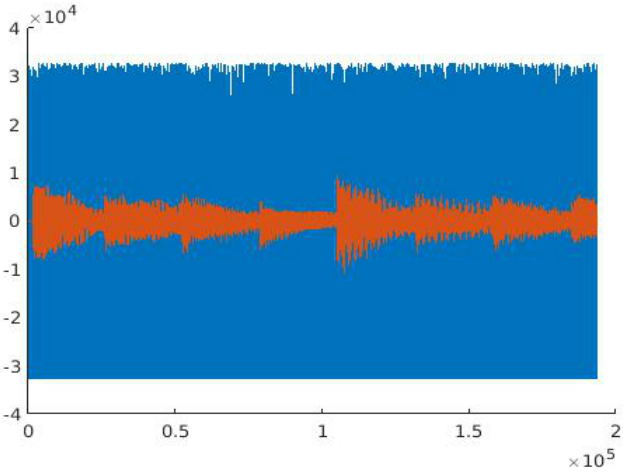


Fig. 8. Difference between the original and transmitted audio signal with 4-Sets-Cut-Decoding for $p = 0.17$

In Table 1 and Table 2, the results for the values of *BER* and *PER* are summarized. From these tables, same as from the previous figures, one can see that 4-Sets-Cut-Decoding algorithm is better than Cut-Decoding algorithm.

TABLE 1: *PER* FOR BOTH ALGORITHMS

| $p$ | $PER_{cut\text{-}decoding}$ | $PER_{4\text{-}sets\text{-}cut\text{-}decoding}$ |
|---|---|---|
| 0.05 | 0.001697359 | 0.000581287 |
| 0.08 | 0.023111979 | 0.007091704 |
| 0.11 | 0.113071987 | 0.034249442 |
| 0.14 | / | 0.129580543 |
| 0.17 | / | 0.34749349 |

TABLE 2: *BER* FOR BOTH ALGORITHMS

| $P$ | $BER_{cut\text{-}decoding}$ | $BER_{4\text{-}sets\text{-}cut\text{-}decoding}$ |
|---|---|---|
| 0.05 | 0.000913267 | 0.000260610 |
| 0.08 | 0.011926399 | 0.003033027 |
| 0.11 | 0.057805137 | 0.015375047 |
| 0.14 | / | 0.056156219 |
| 0.17 | / | 0.347493490 |

All audio files transmitted through the noise channel with different bit-error probability can be found on the link https://www.dropbox.com/sh/mt36x7rq1u5czqu/AAC0zcKiODy4fYOWoTNx6cmGa?dl=0. If one listen these audio files, he/she can notice the following: as $p$ increases, the noise increases too, but the original melody can be listened completely in background.

## III. CONCLUSION

In this paper we investigate the performances of Cut-Decoding and 4-Sets-Cut-Decoding algorithms defined for RCBQ for transmission of audio files. For that aim, we present and compare several experimental results obtained for different values of bit-error probability $p$ in binary-symmetric channel.

From the results we can conclude that for all values of $p$, 4-Sets-Cut-Decoding algorithm gives better results than Cut-Decoding algorithm. Also, 4-Sets-Cut-Decoding algorithm is from 1.2 to 6.2 times faster than Cut-Decoding algorithm.

## REFERENCES


[1] H. Tzonelih, T.R.N. Rao, "Secret error-correcting codes," in *Advances in Cryptology - CRYPTO '88, LNCS 403,* R. Goldwasser, Eds., 1990, pp. 540-563.

[2] N. Zivic, C. Ruland, "Parallel Joint Channel Coding and Cryptography," *International Journal of Electrical and Electronics Engineering*, vol. 4, no. 2, pp.140-144, 2010.

[3] C.N. Mathur, K. Narayan, K.P. Subbalakshmi, "High Diffusion Cipher: Encryption and Error Correction in a Single Cryptographic Primitive," *Applied Cryptography and Network Security 2006, LNCS 3989*, pp.309-324.

[4] D. Gligoroski, S. Markovski, Lj. Kocarev, "Error-correcting codes based on quasigroups," in *Proc. 16th International Conference on Computer Communications and Networks* (ICCCN 2007), 2007, pp.165-172.

[5] D. Gligoroski, S. Markovski, Lj. Kocarev, "Totally asynchronous stream ciphers + Redundancy = Cryptcoding," in *Proc. of the 2007 International Conference on Security and management, SAM 2007* S. Aissi, H.R. Arabnia , Eds., CSREA Press, Las Vegas, 2007, pp. 446 – 451.

[6] A. Popovska-Mitrovikj, S. Markovski, V. Bakeva, "Performances of error-correcting codes based on quasigroups," in *ICT-Innovations 2009*, D. Davcev, J.M. Gomez, Eds., Springer, 2009, pp. 377-389.

[7] A. Popovska-Mitrovikj, S. Markovski, V. Bakeva, "Increasing the decoding speed of random codes based on quasigroups," in *Web Proc. ICT Innovations2012*, ISSN 1857-7288, Ohrid, 2012, pp. 93-102.

[8] A. Popovska-Mitrovikj, S. Markovski, V. Bakeva, "4-Sets-Cut-Decoding algorithms for random codes based on quasigroups," *International Journal of Electronics and Communications* (*AEÜ*), vol. 69, pp. 1417-1428, October 2015.