Journal of Engineering Science and Technology Review
Special Issue on Telecommunications, Informatics, Energy and Management 2019

JOURNAL OF
Engineering Science and Technology Review

Conference Article

www.jestr.org

Cryptcoding of Images for Transmission Trough a Burst Channels

Daniela Mechkaroska, Aleksandra Popovska-Mitrovikj* and Verica Bakeva

Faculty of Computer Science and Engineering, Ss. Cyril and Methodius University, Skopje, Republic of North Macedonia

Received 25 September 2019; Accepted 24 February 2020

Abstract

Random Codes Based on Quasigroups (RCBQ) are error-correcting codes that crypt the messages at the same time. These cryptcodes are proposed in 2007 and after that several improvements of coding/decoding algorithms have been made. For better performances for transmission through a binary-symmetric and Gaussian channel, Cut-Decoding and 4-Sets-Cut-Decoding algorithms were defined. Also, a modification of these algorithms (Burst-Cut-Decoding and Burst-4-Sets-Cut-Decoding algorithms) for correction of burst errors was proposed elsewhere. In this paper, we investigate performances of these algorithms for transmission of images through a burst channel. For simulation of burst errors, we use Gilbert-Elliot model. We consider two kinds of Gilbert-Eliott channel, in the first one in each state the channel is binary symmetric and in the second one, the channel is Gaussian. In all experiments, for different values of bit-error probability (in BSC) and *SNR* (in Gaussian), the differences between transmitted and decoded images are considered. From the experiments can be concluded that Burst-4-Sets-Cut-Decoding algorithms gives better results than Burst-Cut-Decoding algorithms (i.e., clearer images) and it is much faster. Also, a filter is applied on the images (after decoding with RCBQs) for enhancing the quality of them. With the considered filter clearer images are obtained. In this paper we consider only error-correction capabilities of RCBQs, but the images decoded with these codes are also encrypted.

Keywords: Cryptcoding; Gilbert-Elliott channel; SNR; bit-error probability; burst errors; image.

1. Introduction

Random Codes Based on Quasigroups (RCBQ) defined in [1] are error-correcting codes that encrypt the messages, i.e., they are cryptcodes. There are several coding/decoding algorithms for RCBQ for transmission through a binary symmetric and Gaussian channel: Standard algorithm [1], Cut-Decoding algorithm [2] and 4-Sets-Cut-Decoding algorithm [3]. In the process of coding/decoding of these codes, an encryption/decryption algorithm is used and therefore these error-correcting codes encrypt the messages. A few similar combinations of error-correcting codes and cryptographic algorithms are proposed for cryptographic purposes in [4], [5] and [6]. Cryptographic properties of the transformations used in RCBQ are already investigated in several papers [7], [8], [9]. Here, we consider only error-correction capabilities of RCBQs.

In order to improve performances of these codes for correction of burst errors, in [10] authors proposed a new coding/decoding algorithms called Burst-Cut-Decoding and Burst-4-Sets-Cut-Decoding algorithm. In this paper we consider performances of these algorithms for decoding images transmitted through a burst channel simulated with a Gilbert-Elliot model. We consider two kinds of Gilbert-Eliott channels, in the first one, in each state the channel is binary symmetric and in the second one, in each state the channel is Gaussian. In all experiments, for different values of bit-error probability (in BSC) or different values of *SNR* (in Gaussian channel) the differences between transmitted and decoded images are considered. Also, in order to enhance the quality of decoded images a filter is applied on the images after decoding with RCBQs.

2. Coding/decoding algorithms for RCBQ

In this section we will briefly explain coding/decoding algorithms for RCBQs. A detailed description of the algorithms is given in previous papers for these codes [2], [3].

Coding/decoding algorithms of RCBQs are designed using algorithms for encryption/decryption from the implementation of TASC (Totally Asynchronous Stream Ciphers) by quasigroup string transformation ([11]). These cryptographic algorithms use the alphabet Q and a quasigroup operation * on Q together with its parastrophe /. In our experiments we use the alphabet of 4-bit symbols (nibbles) and the quasigroup given in [3].

RCBQs are firstly proposed in [1] and coding/decoding algorithms given there we will denote as Standard coding/decoding algorithm. The previous investigations of RCBQ ([12]) showed that the speed of the decoding process is the biggest problem for Standard RCBQs. The main reason for this is the length of the lists (called decoding-candidate sets), since the decoding of RCBQs is actually a list decoding. The length of the lists depends on the parameter B_{max} - the predicted maximal number of bit errors

* E-mail address: aleksandra.popovska.mitrovikj@finki.ukim.mk
ISSN: 1791-2377

appeared during transmission of a block. The larger value of B_{max} gives larger lists and slower decoding process. In order to improve the decoding speed, in [2] and [3] authors proposed two new coding/decoding algorithms: Cut-Decoding and 4-Sets-Cut-Decoding algorithms. In these algorithms, coding/decoding of the messages is in two (in Cut-Decoding) or four (in 4-Sets-Cut-Decoding) parallel processes. The decoding-candidate sets are reduced using intersection of the sets obtained in the parallel decoding processes. On this way, faster decoding and better results for packet-error (*PER*) and bit-error (*BER*) probabilities are obtained.

The decoding rules in both algorithms (Cut-Decoding and 4-Sets-Cut-Decoding) are the following. After the last iteration, if there is only one element in the list (all reduced sets have only one element with same second component) then this element is the decoded message. In this case, we say that we have a *successful decoding*. If the decoded message is not the correct one then we have an *undetected-error*. If the length of the list in the last iteration is greater than 1 (the reduced sets have more than one element) then we have a *more-candidate-error*. In this case we apply a heuristic: we randomly select a message from the reduced sets in the last iteration and we take this message as the decoded message. If in some iteration all decoding-candidate sets are empty then the process will be stopped (a *null-error* appears). But, if we obtain at least one nonempty decoding-candidate set then the decoding continues with the nonempty sets (the reduced sets are obtained by intersection of the nonempty sets only). If we obtain only one nonempty set, in some iteration, then the decoding continues with the nonempty set using the Standard RCBQ decoding algorithm.

In all decoding algorithms for RCBQ, when a *null-error* appears, the decoding process ends earlier and only a part of the message is decoded. Therefore, in the experiments with images we use the following solution. In the cases when a *null-error* appears we take the strings without redundant symbols from all elements in the sets from the previous iteration and we find their maximal common prefix substring. If this substring has k symbols, then in order to obtain decoded message of l symbols we take these k symbols and we add $l - k$ zero symbols at the end of the message.

For transmission through a burst channels, Cut-Decoding and 4-Sets-Cut-Decoding algorithms do not give good results. Therefore, in [10] authors propose new algorithms for coding/decoding called Burst-Cut-Decoding and Burst-4-Sets-Cut-Decoding algorithms. It is known that interleaving and deinterleaving are useful for handling burst errors in a communication system. So, in these algorithms, we include an interleaver in coding algorithm and the corresponding deinterleaver in the decoding algorithm. Namely, in the process of coding before the concatenation of two (or four) codewords we apply the interleaving on each codeword, separately. The interleaver rearranges (by rows) m nibbles of a codeword in a matrix of order $(m/k) \times k$. The output of the interleaver is a mixed message obtained reading the matrix by columns. Then, after transmission of a concatenated message through a burst channel we divide the outgoing message in two (or four) messages with equal length and before the parallel decoding we apply deinterleaving on each message, separately. In this way, better results for packet-errors and bit-error probabilities for burst channel transmission are obtained.

3. Experiments

In this section we present experimental results obtained with RCBQ for transmission of images through a burst channel. Namely, we investigate performances of Burst-Cut-Decoding and Burst-4-Sets-Cut-Decoding algorithms for transmission of images through a Gilbert-Elliot channel.

We made experiments with two kinds of Gilbert-Elliott channels. In the first one, in each state the channel is binary symmetric with bit error probabilities P_e(G) in a good state and P_e(B) in a bad state. In the second one, two channels are Gaussian where SNR_G in a good state is high and SNR_B in a bad state is low.

All experiments, presented in this paper, are made for code (72, 576) with rate R=1/8, B_{max} = 5 and the following parameters:

- In Burst-Cut-Decoding algorithm - redundancy pattern: 1100 1100 1000 0000 1100 1000 1000 0000 1100 1100 1000 0000 1100 1000 1000 0000 0000 0000, for rate 1/4 and two different keys of 10 nibbles.
- In Burst-4-Sets-Cut-Decoding algorithm – redundancy pattern: 1100 1110 1100 1100 1110 1100 1100 1100 0000 for rate 1/2 and four different keys of 10 nibbles.
- In all experiments we used the same quasigroup on Q given in [3].

In our experiments we use the image of "Lenna" given in Fig. 1.

Fig. 1. Image of "Lenna"

In order to visually enhance damaged pixels and improve the image, in [13] we define a filter that transforms pixel intensity values of the pixels damaged by both types of detected errors (*null-errors* and *more-candidate-errors*). One pixel is considered as damaged if it belongs in a zero sub-block with at least four consecutive zero nibbles. For detection of *more-candidate-errors* in the filter, before applying the filter, we replace the decoded message (randomly chosen using heuristic) with a zero message. The basic idea in the definition of this filter is to replace damaged pixel intensity value with a new value taken over a neighborhood of fixed size. In this process we use the median of the nonzero gray values of the surrounding pixels, so the filter is a median one. In our experiments we compare the images obtained without and with the filter.

First, in Fig. 2 we present some images obtained after transmission through the channel without using any error-correcting code. The first one is obtained after transmission

through a Gilbert-Elliott with BSCs and the second one − with Gaussian channels.

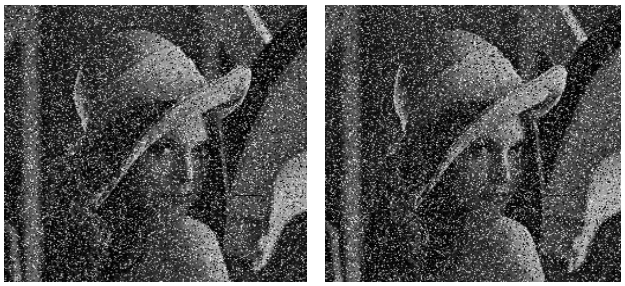


Fig. 2. Images obtained after transmission through the channel without using any error-correcting code

3.1. Experiments for Gilbert-Elliott with BSC channels

In all experiments for Gilbert-Elliott model with binary symmetric channels, we use bit-error probability in the good state $P_e(G) = 0.01$ and a few different values of bit-error probabilities in the bad state $P_e(B) \in \{0.16, 0.13, 0.1\}$ and the following combinations of transition probabilities from good to good state P_{GG} and from bad to bad state P_{BB}:

- $P_{GG} = 0.2$ and $P_{BB} = 0.8$
- $P_{GG} = 0.8$ and $P_{BB} = 0.2$

In Fig. 3a) we present the images for $P_{GG} = 0.2$, $P_{BB} = 0.8$ and all considered $P_e(B)$ obtained with Burst-Cut-Decoding algorithm and in Fig. 3b) the corresponding images obtained after applying the proposed filter. Images obtained for $P_{GG} = 0.8$ and $P_{BB} = 0.2$ are given in Fig. 4.

$P_e(B) = 0.16$ $P_e(B) = 0.13$ $P_e(B) = 0.1$

a) Images without a filter

$P_e(B) = 0.16$ $P_e(B) = 0.13$ $P_e(B) = 0.1$

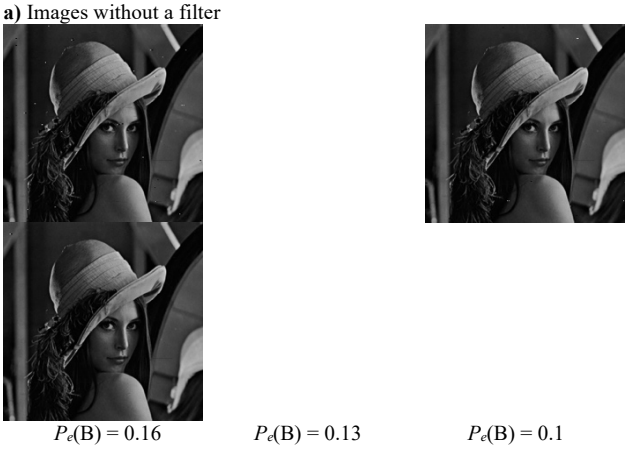
b) Images with a filter

Fig. 3. Images obtained with Burst-Cut-Decoding algorithm for $P_{GG} = 0.2$ and $P_{BB} = 0.8$

From the images in Fig. 3a) and Fig. 5a) (for $P_{GG} = 0.2$ and $P_{BB} = 0.8$) we can conclude that Burst-4-Sets-Cut-Decoding algorithm gives better results than Burst-Cut-Decoding algorithm for all considered values of $P_e(B)$. Comparing the images before and after applying the filter we can conclude that the filter significantly enhances the quality of images decoded with both algorithms. But, it is visible that the filter gives better results for the images obtained with Burst-Cut-Decoding algorithm than with Burst-4-Sets-Cut-Decoding algorithm. The reason for this is the larger number of *undetected-errors* in the experiments with Burst-4-Sets-Cut-Decoding algorithm. Namely, the filter cannot detect this kind of errors.

$P_e(B) = 0.16$ $P_e(B) = 0.13$ $P_e(B) = 0.1$

a) Images without a filter

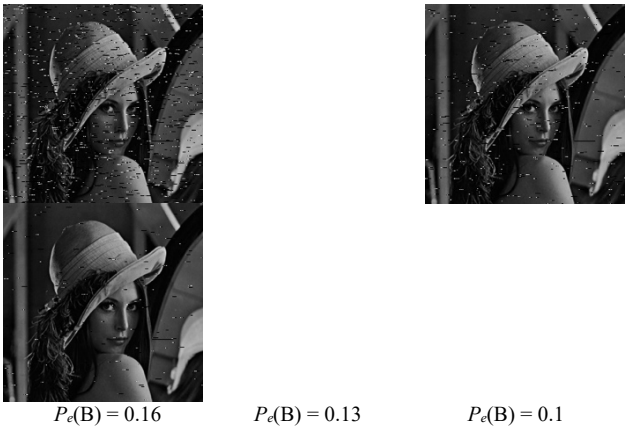


$P_e(B) = 0.16$ $P_e(B) = 0.13$ $P_e(B) = 0.1$

b) Images with a filter

Fig. 4. Images obtained with Burst-Cut-Decoding algorithm for $P_{GG} = 0.8$ and $P_{BB} = 0.2$

Images obtained with Burst-4-Sets-Cut-Decoding algorithm for $P_{GG} = 0.2$, $P_{BB} = 0.8$ are presented in Fig. 5 and for $P_{GG} = 0.8$ and $P_{BB} = 0.2$ in Fig. 6.



$P_e(B) = 0.16$ $P_e(B) = 0.13$ $P_e(B) = 0.1$

a) Images without a filter

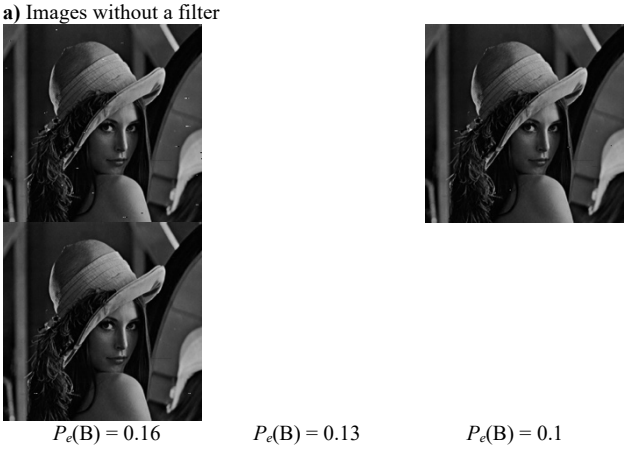
P_e(B) = 0.16 P_e(B) = 0.13 P_e(B) = 0.1

b) Images with a filter

Fig. 5. Images obtained with Burst-4-Sets-Cut-Decoding algorithm for $P_{GG} = 0.2$ and $P_{BB} = 0.8$

P_e(B) = 0.16 P_e(B) = 0.13 P_e(B) = 0.1

a) Images without a filter



P_e(B) = 0.16 P_e(B) = 0.13 P_e(B) = 0.1

b) Images with a filter

Fig. 6. Images obtained with Burst-4-Sets-Cut-Decoding algorithm for $P_{GG} = 0.8$ and $P_{BB} = 0.2$

The images for $P_{GG} = 0.8$ and $P_{BB} = 0.2$ (in Fig. 4 and Fig. 6) are clearer due to the smaller number of errors in the channels with these transition probabilities. Therefore, in these images there is no great difference between the images decoded with both algorithms and after applying the filter.

3.2. Experiments for Gilbert-Elliott with Gaussian channels

In this subsection, we present the experimental results for Gilbert-Elliott model with Gaussian channels for $SNR_G = 4$ and different values of $SNR_B \in \{-3, -2, -1\}$. For these channels we made experiments with the same transition probabilities as in the experiments with binary symmetric channels.

Images obtained with Burst-Cut-Decoding algorithm for $P_{GG} = 0.2$, $P_{BB} = 0.8$ and all considered SNR_B are given in Fig. 7a). The images in Fig. 7b) are obtained from the images given in Fig. 7a) after applying the filter. Images for $P_{GG} = 0.8$ and $P_{BB} = 0.2$ are given in Fig. 8.

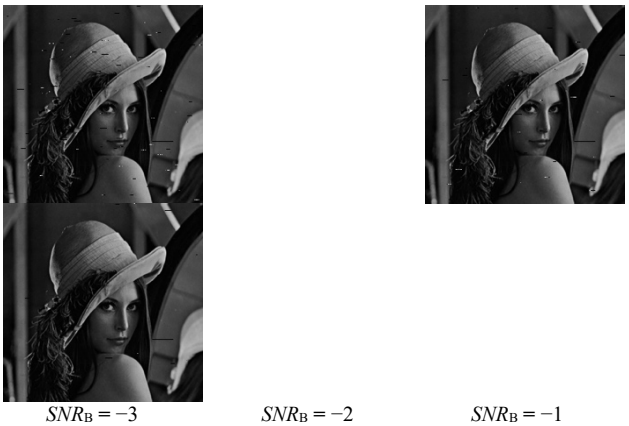
$SNR_B = -3$ $SNR_B = -2$ $SNR_B = -1$

a) Images without a filter

$SNR_B = -3$ $SNR_B = -2$ $SNR_B = -1$

b) Images with a filter

Fig. 7. Images obtained with Burst-Cut-Decoding algorithm for $P_{GG} = 0.2$ and $P_{BB} = 0.8$



$SNR_B = -3$ $SNR_B = -2$ $SNR_B = -1$

a) Images without a filter

$SNR_B = -3$ $SNR_B = -2$ $SNR_B = -1$

b) Images with a filter

Fig. 8. Images obtained with Burst-Cut-Decoding algorithm for $P_{GG} = 0.8$ and $P_{BB} = 0.2$

In Fig. 9a) (without filter) and Fig. 9b) (with the filter) we present the images obtained with Burst-4-Sets-Cut-Decoding algorithm for $P_{GG} = 0.2$ and $P_{BB} = 0.8$. Images obtained with this algorithm for $P_{GG} = 0.8$ and $P_{BB} = 0.2$ are given in Fig. 10.

Analyzing the images given in this subsection (transmitted through a Gilbert-Elliott with Gaussian channels) we can derive the same conclusions as for images transmitted through a Gilbert-Elliott with BSCs.

$SNR_B = -3$ $SNR_B = -2$ $SNR_B = -1$

a) Images without a filter

$SNR_B = -3$ $SNR_B = -2$ $SNR_B = -1$

b) Images with a filter

Fig. 9. Images obtained with Burst-4-Sets-Cut-Decoding algorithm for $P_{GG} = 0.2$ and $P_{BB} = 0.8$

$SNR_B = -3$ $SNR_B = -2$ $SNR_B = -1$

a) Images without a filter

$SNR_B = -3$ $SNR_B = -2$ $SNR_B = -1$

b) Images with a filter

Fig. 10. Images obtained with Burst-4Sets-Cut-Decoding algorithm for $P_{GG} = 0.8$ and $P_{BB} = 0.2$

4. Conclusions

In this paper, we investigate performances of Burst-Cut-Decoding and Burst-4-Sets-Cut-Decoding algorithms for transmission of images through a Gilbert-Elliot channel. From the presented experimental results, we concluded that these algorithms show good performances for all considered channel parameters. We can see that in all experiments a great number of transmission errors is corrected. Also, comparing the images decoded by two considered algorithms we can conclude that Burst-4-Sets-Cut-Decoding algorithms gives better results than Burst-Cut-Decoding algorithms. Comparing the running times, it is evidently that the decoding with Burst-4-Sets-Cut-Decoding algorithm is much faster than decoding with Burst-Cut-Decoding algorithm. Also, the median filter is applied on the images (after decoding with RCBQs) for enhancing the quality of them. On this way, we obtain clearer images for both algorithms and all channel parameters. At the end, we must note that the encryption algorithm used in the design of these codes provides information security of the transmitted images.

Acknowledgements

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

References

1. D. Gligoroski, S. Markovski, and Lj. Kocarev, "Error-correcting codes based on quasigroups", Proceedings of 16th International Conference on Computer Communications and Networks, 2007, pp. 165-172.
2. A. Popovska-Mitrovikj, S. Markovski, and V. Bakeva, "Increasing the decoding speed of random codes based on quasigroups", Markovski, S., Gusev, M. (eds.) ICT Innovations 2012, Web proceedings, ISSN 1857-7288, 2012, pp. 93-102.
3. A. Popovska-Mitrovikj, S. Markovski, and V. Bakeva, "4-Sets-Cut-Decoding algorithms for random codes based on quasigroups", International Journal of Electronics and Communications (AEU), Vol.69(10), Elsevier, 2015, pp. 1417-1428.
4. C. N. Mathur, K. Narayan, and K.P. Subbalakshmi "High Diffusion Cipher: Encryption and Error Correction in a Single Cryptographic Primitive", Applied Cryptography and Network Security 2006, LNCS 3989, 2006, pp. 309-324.
5. H. Tzonelih, and T.R.N Rao, "Secret error-correcting codes", Goldwasser R., (eds.) Advances in Cryptology - CRYPTO 88, LNCS, 403, 1990, pp. 540-563.
6. N. Zivic, and C. Ruland C, "Parallel Joint Channel Coding and Cryptography", International Journal of Electrical and Electronics Engineering, 4(2), 2010, pp.140-144.
7. V. Dimitrova, and J. Markovski, "On Quasigroup Pseudo Random Sequence Generators", Proceedings of 1st Balkan Conference in Informatics, Thessaloniki, Greece, 2003, pp.393-401.
8. S. Markovski, D. Gligoroski, and V. Bakeva, "Quasigroup String Processing: Part 1", Contributions, Section of Natural, Mathematical and Biotechnical Sciences, 20(1-2), 1999, pp. 13-28.
9. S. Markovski, D. Gligoroski, and Lj. Kocarev, "Unbiased Random Sequences from Quasigroup String Transformations", Gilbert, H., Handschuh, H. (eds.): Fast Software Encryption, Springer Berlin Heidelberg, 2005, pp. 163-180.
10. D. Mechkaroska, A. Popovska-Mitrovikj, and V. Bakeva, "New cryptcodes for burst channels", M. Ćirić, M. Droste, J.E.Pin (eds.), Algebraic Informatics, 8th International Conference, CAI 2019, Proceedings, LNCS, Springer, (in print).
11. D. Gligoroski, S. Markovski, and Lj. Kocarev, "Totally asynchronous stream ciphers +Redundancy = Cryptcoding", Aissi, S., Arabnia, H.R. (eds.), Proceedings of the International Conference on Security and management, SAM 2007, CSREA Press, Las Vegas, 2007, pp. 446-451.
12. A. Popovska-Mitrovikj, S. Markovski, and V. Bakeva, "Performances of error-correcting codes based on quasigroups", D. Davcev, J.M. Gomez (eds.), ICT-Innovations 2009, Springer, pp. 377-389.
13. D. Mechkaroska, A. Popovska-Mitrovikj, V. Bakeva, "A filter for Images Decoded using Cryptocodes Based on Quasigroups", Proceedings of the 14th International Conference on Informatics and Information, Mavrovo, April 2017, pp. 52-56.