IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 25 April 2023, accepted 15 May 2023, date of publication 19 May 2023, date of current version 31 May 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3278051

RESEARCH ARTICLE

Fast Decoding of Images With Cryptcodes for Burst Channels

ALEKSANDRA POPOVSKA-MITROVIKJ[1], DANIELA MECHKAROSKA[2], AND VERICA BAKEVA[1]

[1]Faculty of Computer Science and Engineering, Ss. Cyril and Methodius University in Skopje, 1000 Skopje, Macedonia
[2]Faculty of Communication Networks and Security, University of Information Science and Technology "St. Paul the Apostle," 6000 Ohrid, Macedonia

Corresponding author: Aleksandra Popovska-Mitrovikj (aleksandra.popovska.mitrovikj@finki.ukim.mk)

This work was supported in part by the Faculty of Computer Science and Engineering, Ss Cyril and Methodius University in Skopje.

ABSTRACT The concept of cryptcoding arises from the need to obtain secure and accurate transmission. This has led to an intensive development of coding theory and cryptography as scientific fields dealing with these problems. To ensure efficient and secure data transmission at the same time, the concept of cryptcoding is being developed, in which the coding and encryption processes are merged into one process. Cryptcodes provide correction of a certain number of errors in the transmitted message and data confidentiality, using only one algorithm. In this paper, we consider cryptcodes based on quasigroups, proposed elsewhere. Also, Burst-Cut-Decoding, Burst-4-Sets-Cut-Decoding, FastB-Cut-Decoding and FastB-4-Sets-Cut-Decoding algorithms for transmission of messages through burst channels have been defined and investigated elsewhere. Here, we investigate performances of these algorithms for transmission of images through burst channels. We made experiments for different channel parameters and compared the results obtained with different decoding algorithms of these cryptcodes. In all experiments, we considered the bit-error probability and the differences between the transmitted and decoded images. From the results presented in this paper we can conclude that Fast algorithms improve the performances of these cryptcodes for transmission of images over a burst channel. The best results are obtained by FastB-4-Sets-Cut-Decoding algorithm. In addition, to enhance the quality of the decoded images, we examined the application of a filter for visual correction of unsuccessfully decoded pixels using the surrounding pixels. The considered cryptcodes and presented results can be useful for application in satellite digital video broadcasting (DVB-S) coding and encryption schemes.

INDEX TERMS Cryptcoding, error-correcting codes, burst channel, image, quasigroups.

I. INTRODUCTION

Nowadays, when data are transmitted through a noisy channel, in addition to correcting the errors in the messages caused by the noise in the channel, increasing emphasis is placed on the security of the transmitted messages. Thus, there is a need to combine two scientific fields, cryptography and coding theory, which deal with these two problems in data transmission into a concept called cryptcoding.

One way, to ensure secure and accurate transmission, is to use two different algorithms, one for encryption of the message and another for coding [1], [2]. In [3], authors propose a combined scheme of Turbo codes codes and AES where a turbo encoder block is embedded in AES encryption block in the first round. Another secure channel coding scheme based on turbo codes [4], is defined in such a way that the redundant information used for error correction can be selected at random from the entire set of potential strings. In [5] authors proposed a cryptocoding system for flash memory or transmission, where they first encrypt the information using AES, and then encode the encrypted data by concatenated BCH and QC-LDPC codes. Here, we consider error-correcting codes resistant to an intruder attack which consists of only one algorithm that provides a correction of a certain amount of transmission errors and information security. These cryptcodes, called Random Codes Based on Quasigroups are proposed in [6] and they use a cryptographic algorithm during the encoding/decoding process.

The associate editor coordinating the review of this manuscript and approving it for publication was Tony Thomas.

RCBQs use several parameters (redundancy pattern, initial keys, quasigroups) in their design. Their error-correction performances and decoding speed depend on these parameters. The decoding process is a list decoding, and the size of the lists (called decoding candidate sets) has an impact on the decoding speed and probability of correct decoding. Therefore, several modifications of coding/decoding algorithms (Cut-Decoding algorithms, 4-Sets-Cut-Decoding algorithms) are proposed in [7] and [8]. In these algorithms, using intersections of decoding candidate sets obtained in parallel decoding process, a significant speed-up of decoding process is obtained. For improving the performances of RCBQ for transmission through a burst channel, two new algorithms called Burst-Cut-Decoding and Burst-4-Sets-Cut-Decoding algorithm are proposed in [9]. In these algorithms, an interleaver in coding algorithm and the corresponding deinterleaver in the decoding algorithm, are included. In this way, the results for packet-errors and bit-error probabilities are better than with the old Cut-Decoding and 4-Sets-Cut-Decoding algorithm. Also, Burst-4-Sets-Cut-Decoding algorithm gives from 2 to 8 times better results than Burst-Cut-Decoding algorithm [9]. Additionally, to provide faster and more efficient decoding, particularly for transmission over a low-noise channels in [10] and [11] authors consider new modifications (Fast-Cut-Decoding, Fast-4-Sets-Cut-Decoding, FastB-Cut-Decoding and FastB-4-Sets-Cut-Decoding algorithms) of previously mentioned coding/decoding algorithms for RCBQs. In [10] it is concluded that Fast-Cut-Decoding and Fast-4-Sets-Cut-Decoding algorithms provide more efficient and faster decoding, especially for transmission through a low noise Gaussian channel. Also, the results given in [11] show that with FastB-Cut-Decoding and FastB-4-Sets-Cut-Decoding algorithms results for packet-error and bit-error probabilities are better than with Burst-Cut-Decoding and Burst-4-Sets-Cut-Decoding algorithms for transmission through a burst channels. Also, in [11] authors conclude that for some channel parameters FastB-Cut-Decoding and FastB-4-Sets-Cut-Decoding algorithms provide faster decoding.

In this paper, we will examine performances of Burst and FastB algorithms for transmission of images. We made experiments with Burst-Cut-Decoding, Burst-4-Sets-Cut-Decoding, FastB-Cut-Decoding and FastB-4-Sets-Cut-Decoding algorithms for transmission of images through a burst channel, and we compare the obtained results. We will present experimental results for different channel parameters and compare the results obtained with different coding/decoding algorithms of these cryptcodes. In all experiments, we consider the differences between transmitted and decoded images. Also, for enhancing the quality of decoded images, we examine the application of a median filter for visual correction of unsuccessfully decoded pixels using the surrounding pixels. From the presented results it can be seen that Fast algorithms improve the performances of these cryptcodes for transmission of images over a burst channel. Also, the proposed filter enhances the quality of the images for all considered channel parameters.

The rest of the paper is organized as follows. First, we briefly describe coding/decoding process of RCBQs using different (mentioned above) algorithms. In Section III, we explain the Gilbert-Elliot model for burst channel that we use in our experiments and we define the filter for enhancing decoded images. Then, in the same section, we present, analyze and compare the experimental results for image transmission through a burst channel using Burst-Cut-Decoding, Burst-4-Sets-Cut-Decoding algorithm and their fast modifications. At the end, we will derive some conclusions about the practical use of these cryptcodes in the considered noisy channels.

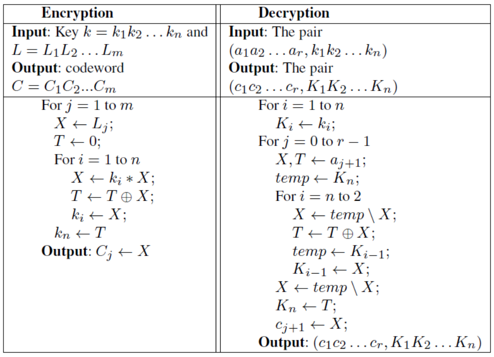
II. CRYPTCODES BASED ON QUASIGROUPS

As we mention in the previous section, RCBQs are cryptcodes that use a cryptographic algorithm in the encoding/decoding process. In the encoding process we use encryption algorithm and in the decoding process - corresponding decryption algorithm from the Totally Asynchronous Stream Ciphers (TASC) designed using quasigroup string transformations [12]. These encryption/decryption algorithms use the alphabet Q and a quasigroup operation $*$ on Q together with its parastrophe $\backslash$. In the experiments for this paper, we use the alphabet of nibbles, the quasigroup of order 16 and its parastrophe, given in [8].

A. ENCODING

In the coding process of the first version of coding/decoding algorithm for RSBQs called Standard algorithm of RCBQ ([6]) first we choose a pattern for adding redundant zero symbols and using this pattern we extend the message M of l nibbles to message L of m nibbles. Then we divide the message L in a blocks of r nibbles. The codeword C for M is obtained by applying the encryption algorithm of TASC (given in Fig. 1) to the message L. For this we need to choose an initial key, k of n nibbles, that is also used in the decoding process. As we can see from the algorithms given in Fig. 1, the complexity of encryption/decryption algorithms is polynomial and depends of the key size.

In Cut-Decoding (Burst-Cut-Decoding, FastB-Cut-Decoding) algorithm, for code with rate R instead of using a pattern for redundant symbols for rate R, we use a two times shorter pattern, i.e., a pattern for rate $2R$. We form the redundant message L and we apply the encryption algorithm (given in Fig. 1) two times, but using different parameters, different keys or different quasigroups, and we obtain two parts of the codeword. The concatenation of these two parts give the codeword of the message. Similarly, in 4-Sets-Cut-Decoding (Burst-4-Sets-Cut-Decoding, FastB-4-Sets-Cut-Decoding) algorithm we use a four times shorter pattern (for code with rate $4R$) and we apply the encryption algorithm four times to obtain 4 parts of the codeword.



```
Encryption
Input: Key $k = k_1k_2 \ldots k_n$ and
$L = L_1L_2 \ldots L_m$
Output: codeword
$C = C_1C_2 ... C_m$
    For $j = 1$ to $m$
        $X \leftarrow L_j$;
        $T \leftarrow 0$;
        For $i = 1$ to $n$
            $X \leftarrow k_i * X$;
            $T \leftarrow T \oplus X$;
            $k_i \leftarrow X$;
        $k_n \leftarrow T$
    Output: $C_j \leftarrow X$
```

```
Decryption
Input: The pair
$(a_1a_2 \ldots a_r, k_1k_2 \ldots k_n)$
Output: The pair
$(c_1c_2 \ldots c_r, K_1K_2 \ldots K_n)$
    For $i = 1$ to $n$
        $K_i \leftarrow k_i$;
    For $j = 0$ to $r - 1$
        $X, T \leftarrow a_{j+1}$;
        $temp \leftarrow K_n$;
        For $i = n$ to 2
            $X \leftarrow temp \setminus X$;
            $T \leftarrow T \oplus X$;
            $temp \leftarrow K_{i-1}$;
            $K_{i-1} \leftarrow X$;
        $X \leftarrow temp \setminus X$;
        $K_n \leftarrow T$;
        $c_{j+1} \leftarrow X$;
    Output: $(c_1c_2 \ldots c_r, K_1K_2 \ldots K_n)$
```

FIGURE 1. Algorithms for encryption and decryption.

B. DECODING

After transmission through a noisy channel, we receive a message D. In all algorithms, the decoding is iterative. In each iteration we form a decoding candidate set. The speed of the decoding process depends of the cardinality of these sets. For the decoding we need to divide the message D in a blocks of r nibbles and to choose an integer B_{max} - the assumed maximum number of bit transmission errors in one block. In the i^{th} iteration of the decoding process we form the set H_i which contains the strings of r nibbles that are at Hamming's distance $\leq B_{max}$ from the i^{th} block of the message D and we use this set in forming the decoding candidate set S_i of the i^{th} iteration. In all algorithms we start with an initial set $S_0 = (k; \lambda)$, where λ is the empty sequence, and k is the initial key used in the coding process.

In the i^{th} iteration of the Standard decoding process of RCBQs, using the sets S_{i-1} and H_i we construct the set S_i of elements $(\beta, w_1 \ldots w_{ri})$, where w_j are nibbles, on the following way. For each element $h \in H_i$ and each $(\alpha, w_1 \ldots w_{r(i-1)}) \in S_{i-1}$, we apply the decryption algorithm of TASC, given in Fig. 1, with input (h, α). If the output is the pair (δ, β) and if the string δ has redundant zero nibbles at the same positions as the chosen pattern, then $(\beta, w_1 \ldots w_{r(i-1)}c_1c_2 \ldots c_r) \equiv (\beta, w_1 \ldots w_{ri})$ is an element of S_i.

In Cut-Decoding algorithm ([7]), we split the received message D in two messages of equal lengths and we decode them in parallel with the parameters used in two applications of the encryption algorithm in the encoding process. So, in each iteration of the decoding process we form two decoding candidate sets $S_i^{(1)}$ and $S_i^{(2)}$, Before the next iteration we reduce the number of elements in these sets by removing from $S_i^{(1)}$ all elements whose second part does not match with the second part of an element in $S_i^{(2)}$, and vice versa. In the next iteration the both processes use the reduced sets.

In 4-Sets-Cut-Decoding algorithm ([8]) we split the received message D in four messages of equal lengths and we decode them in parallel with the parameters used in the four applications of the encryption algorithm in the encoding process. Similarly, as in Cut-Decoding algorithm, in each iteration we reduce the four obtained decoding candidate sets. Here, we use the following algorithm for reduction. If $S_i^{(1)}$, $S_i^{(2)}$, $S_i^{(3)}$ and $S_i^{(4)}$ are the decoding candidate sets obtained in the i^{th} iteration of the decoding process, then V_1, V_2, V_3 and V_4 are sets of all strings that are second part of an element in $S_i^{(1)}$, $S_i^{(2)}$, $S_i^{(3)}$ and $S_i^{(4)}$, correspondingly, and $V = V_1 \cap V_2 \cap V_3 \cap V_4$. If $V = \emptyset$ then $V = (V_1 \cap V_2 \cap V_3) \cup (V_1 \cap V_2 \cap V_4) \cup (V_1 \cap V_3 \cap V_4) \cup (V_2 \cap V_3 \cap V_4)$. Next, we remove from $S_i^{(1)}$, $S_i^{(2)}$, $S_i^{(3)}$ and $S_i^{(4)}$ all pairs with second part that is not in V. In the next iteration all four parallel decoding process use these smaller decoding candidate sets.

Since, in each iteration we form the decoding candidate sets using the sets in the previous iteration it is clear that if in some iteration all decoding candidate sets are empty the decoding process can not continue. In this case we say that a *null-error* appears. But, in Cut-Decoding and 4-Sets-Cut-Decoding algorithms if we have at least one nonempty decoding candidate set then the decoding continues with the nonempty sets and the reduced sets are obtained by intersection of the non-empty sets only. If after the last iteration the reduced decoding candidate sets have more than one element, then we have a *more-candidate-error*. In this case, we can choose one message from the intersection of the reduced sets and take this message as a decoded message. Experiments shows that almost always the correct message is in the decoding candidate sets. We have a *successful decoding*, if all reduced sets (two in Cut-Decoding algorithm, four in 4-Sets-Cut-Decoding algorithm) in the last iteration have only one element with a same second component. This component is the decoded message L and the messages M is obtained by removing the redundant zero nibbles. If the decoded message is not correct, then we say that an *undetected-error* appears.

For transmission over a binary-symmetric and Gaussian channels Cut-Decoding and 4-Sets-Cut-Decoding algorithms give a great improvement of the performances of RCBQ, compared with the Standard algorithm. But, they do not give good results in the experiments for burst channels. To solve this problem we use an interleaver in the coding process, and the corresponding deinterleaver in the decoding process. Thus, we define algorithms called Burst-Cut-Decoding and Burst-4-Sets-Cut-Decoding algorithms [9]. In encoding process of Burst algorithms, before concatenating two (or four) codewords, we apply interleaving to each codeword, separately. With this, we scatter accumulated burst errors throughout the message. Then, after transmission through a burst channel we split the outgoing message D in two (or four) parts of equal length and we apply deinterleaver on each part, separately. Then parallel decoding of two (four) parts of the message continues using the appropriate decoding algorithm.

As we mentioned previously, the speed of decoding process depends on the cardinality of the decoding candidate sets (smaller sets gives faster decoding). From the algorithm for forming decoding candidate sets it is clear that for smaller values of B_{max} (the assumed maximum number of bit transmission errors in one block), smaller sets are obtained. But, we do not know the number of transmission errors in a block, in advance. If this number of errors in one block is larger than B_{max}, the algorithm will fail to correct the errors. On the other side, if we choose too large value for B_{max}, we will obtain large decoding candidate sets and the decoding process will be too slow. Also, this can lead to a *more-candidate-error*. In this case, if there are no more than B_{max} transmission errors the correct message will be in the decoding candidate sets of the last iteration. This means that even when the bit-error probability of the channel is small (the number of bit errors in a block is not greater than B_{max}), or there are no errors during transmission, *more-candidate-errors* can be obtained. This give us an idea to made a modifications of Cut-Decoding and 4-Sets-Cut-Decoding algorithms, called Fast-Cut-Decoding and Fast-4-Sets-Cut-Decoding algorithms [10]. In this algorithms, instead of using a fixed value B_{max}, we start the decoding process with $B_{max} = 1$. If we have successful decoding, the procedure is done. If not, we increase the value of B_{max} by 1 and we repeat the decoding process with the new value of B_{max}, etc. We finish the decoding with $B_{max} = 4$ (for codes with rate $1/4$) or with $B_{max} = 5$ (for codes with rate $1/8$). In these algorithms we try to decode using smaller decoding candidate sets and in the case of successful decoding with a small value of B_{max} ($B_{max} < 4$), we avoid large sets, and the decoding of the message is faster. Also, on this way we decrease the number of *more-candidate-errors*. In [11] we made this modification of the decoding process in Burst-Cut-Decoding and Burst-4-Sets-Cut-Decoding algorithms, and we proposed so called FastB-Cut-Decoding and FastB-4-Sets-Cut-Decoding algorithms.

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, when we use these codes for transmission of images and a *null-error* appears, we find the maximum common prefix sub-string for all the strings (without redundant zeros) from the previous iteration's decoding candidate sets. If the length of this sub-string is k, then we take those k symbols and add $l-k$ zero symbols at the end of the message to produce the decoded message of l symbols. Also, in order to be able to detect a place of *more-candidate-errors* in an image, when decoding ends with this type of error we take a message of all zero symbols as a decoded message. Using these zero symbols (added in the case of a *null-error* or a *more-candidate-error*) in [13] we propose a filter for enhancing the quality of decoded images. But, with this filter we cannot enhance pixels damaged by *undetected-errors* (we can not locate this type of errors). We use this filter for enhancing the decoded images obtained in the experiments made for this paper.

III. EXPERIMENTS AND ANALYSIS

A. GILBERT-ELLIOTT BURST MODEL

For simulation of burst errors we use Gilbert-Elliott Burst Model introduced by Edgar Gilbert and E. O. Elliott. The basis of this widely used model is a Markov chain with two states: G (good or gap) and B (bad or burst). The probability of a bit being transmitted incorrectly is small in the good state, and large in the bad state.

The model is presented in Fig. 2, where G indicates the good state and B the bad state. The probability of moving from bad to good state is P_{BG} and the probability of moving from good to bad state is P_{GB} [14], [15].

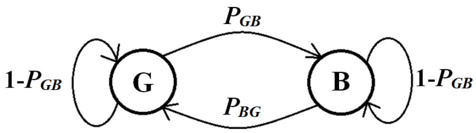


FIGURE 2. Gilbert-Elliott Burst Model.

B. CODE PARAMETERS USED IN THE EXPERIMENTS

Cut-Decoding and 4-Sets-Cut-Decoding algorithms improve performances of RCBQs for transmission through a Gaussian channel and binary-symmetric channel. But, these algorithms do not give good results for burst channels. Therefore in [9] authors propose the modification of these algorithms called Burst-Cut-Decoding and Burst-4-Sets-Cut-Decoding algorithms. In the same paper performances of Cut-Decoding and 4-Sets-Cut-Decoding algorithms are compared with the suitable Burst versions of the algorithms and it is concluded that Burst algorithms give much better results than Cut-Decoding and 4-Sets-Cut-Decoding algorithms when transmission is over a burst channel.

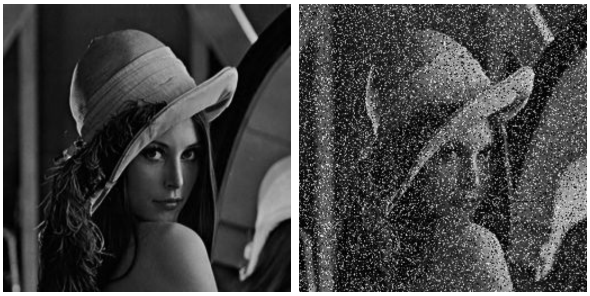
In this paper, our goal is to compare performances of Burst-Cut-Decoding and Burst-4-Sets-Cut-Decoding algorithms with FastB-Cut-Decoding and FastB-4-Sets-Cut-Decoding algorithms, correspondingly. We made experiments for code (72, 576) using different parameters in the coding/decoding process. Here, we will present the best experimental results obtained using the following code parameters:

- in Burst-Cut-Decoding/FastB-Cut-Decoding - redundancy pattern: 1100 1100 1000 00001100 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/FastB-4-Sets-Cut-Decoding - redundancy pattern: 1100 1110 1100 1100 1110 1100 1100 1100 0000 for rate $1/2$ and four different keys of 10 nibbles.

The above redundancy pattern are used for obtaining the message L in the encoding process. Namely, we form extended message L such that on the place of ”1” in the pattern we put a nibble form the original message, and on the place of ”0” we put zero nibble. The same pattern is used in the decoding process for forming the decoding candidate sets, i.e., in the i-th iteration we check whether the output of the decryption algorithm has the redundancy zero nibbles in the same position as the i-th block in the pattern.

The maximum value of B_{max} is 5 and we use the same quasigroup on Q of order 16 and its parastrophe, given in [8].

In all experiments for this paper we use the image of "Lenna", given in Fig. 3 a). The second image in Fig. 3 is the image obtained at the output of a burst channel, without use of any error-correcting code.



a) b)

FIGURE 3. Original and image without using any error-correcting code.

C. FILTER FOR IMAGES DECODED BY CRYPTCODES BASED ON QUASIGROUPS

In order to enhance the quality of the decoded images, we apply a filter. We try a different filters, but the best results are obtained using the following median filter. First, the filter has to identify the location of the damaged pixels. This is easy in the case of *null-error*, because then we add zero symbols in the place of the undecoded part of the message. To enable the location of *more-candidate-errors*, when this type of error appears we take a message of all zero symbols as a decoded message. In the filter, we consider the pixel as damaged if it is in a sub-block with at least four consecutive zero nibbles. The basic concept of this filter is to replace a damaged pixel intensity value with a new value that is a median of the nonzero gray values of the surrounding pixels.

D. EXPERIMENTAL RESULTS

In this subsection, we present experimental results obtained with RCBQ for transmission through a burst channel with considered algorithms. We compare the values of bit-error probability (*BER*) and images obtained with Burst-Cut-Decoding, Burst-4-Sets-Cut-Decoding, FastB-Cut-Decoding and FastB-4-Sets-Cut-Decoding algorithms.

In this subsection, we present the experimental results for Gilbert-Elliott model with Gaussian channels with $SNR_G = 4$ (SNR in good state) and for different values of $SNR_B = -3, -2, -1$ (SNR in bad state). We made experiments with the following two combinations of transition probabilities P_{GG} from good to good state and P_{BB} from bad to bad state:

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

We choose these combinations for transition probabilities to see the different behavior of the channel when the transition probability from bad to bad state is larger than the transition probability from good to good state and opposite. Namely, with the first combination of probabilities we obtain a channel with a bigger probability to be in a good state, which corresponds to low noise. On the other side, the situation with the second combination of the probabilities is opposite, the channel is more often in the bad state with high noise.

The image is transmitted through the channel and the corresponding decoding algorithm is applied.

In Table 1 and Fig. 4- 5, we present experimental results for bit-error probabilities BER_{fb-cut}, BER_{b-cut}, $BER_{fb-4sets}$, $BER_{b-4sets}$ obtained with FastB-Cut-Decoding algorithm, Burst-Cut-Decoding algorithm, FastB-4-Sets-Cut-Decoding algorithm and Burst-4-Sets-Cut-Decoding algorithm, respectively.

We can conclude that with FastB-Cut-Decoding and FastB-4-Sets-Cut-Decoding algorithms, we obtained better results

TABLE 1. Experimental results for *BER*.

SNR_B	BER_{fb-cut}	BER_{b-cut}	$BER_{fb-4sets}$	$BER_{b-4sets}$
		$P_{GG} = 0.2$	$P_{BB} = 0.8$	
−3	0.17161	0.19323	0.06478	0.08006
−2	0.06829	0.08234	0.01663	0.03463
−1	0.01843	0.03461	0.00362	0.01895
		$P_{GG} = 0.8$	$P_{BB} = 0.2$	
−3	0.00200	0.00571	0.00012	0.002258
−2	0.00063	0.00192	0	0.000814
−1	0.00026	0.00065	0	0.000139

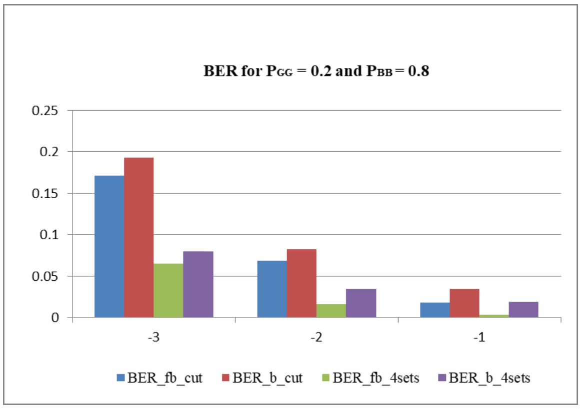


FIGURE 4. Experimental results for *BER* for $P_{GG} = 0.2$ and $P_{BB} = 0.8$.

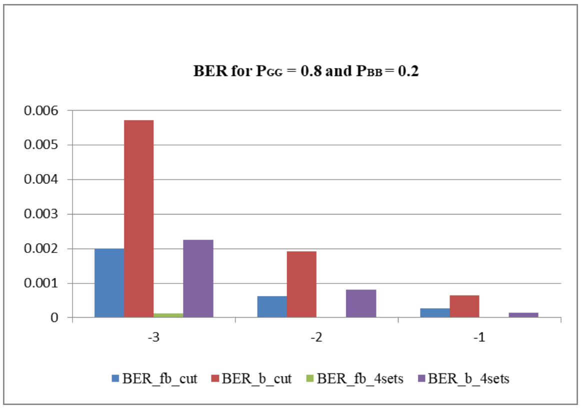


FIGURE 5. Experimental results for *BER* for $P_{GG} = 0.8$ and $P_{BB} = 0.2$.

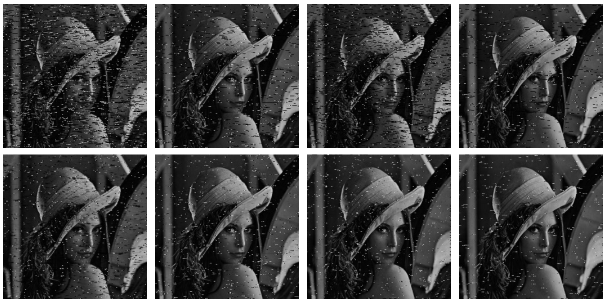


FIGURE 6. Images for $P_{GG} = 0.2$, $P_{BB} = 0.8$ and $SNR = -3$.

for bit-error probabilities than with Burst-Cut-Decoding and Burst-4-Sets-Cut-Decoding algorithms of RCBQ for transmission through a burst channels. Also, we can see that FastB-4-Sets-Cut-Decoding algorithm gives from 2 to 16 times better results than FastB-Cut-Decoding algorithm.

For visual comparison in Fig. 6 – Fig. 8, we present images obtained for $SNR_B = -3$, $SNR_B = -2$, and $SNR_B = -1$, correspondingly, for $P_{GG} = 0.2$, $P_{BB} = 0.8$. In each figure, the first image is obtained using Burst-Cut-Decoding, the second image is obtained using Burst-4-Sets-Cut-Decoding algorithm, the third image using FastB-Cut-Decoding algorithm and the fourth one using FastB-4-Sets-Cut-Decoding algorithm. In the second row, we give the corresponding images obtained after applying the filter defined in Subsection III-C.

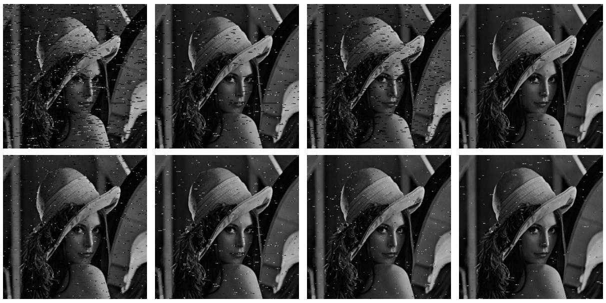


FIGURE 7. Images for $P_{GG} = 0.2$, $P_{BB} = 0.8$ and $SNR = -2$.

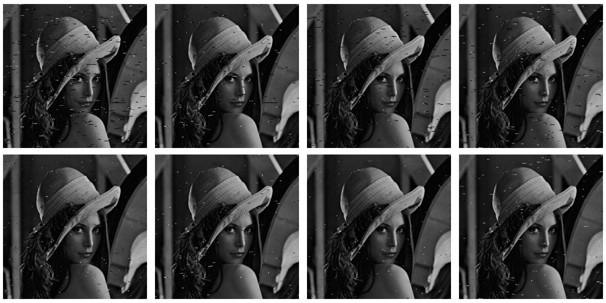


FIGURE 8. Images for $P_{GG} = 0.2$, $P_{BB} = 0.8$ and $SNR = -1$.

In Fig. 9 – Fig. 11, we present images obtained for $SNR_B = -3$, $SNR_B = -2$, and $SNR_B = -1$, correspondingly, for $P_{GG} = 0.8$, $P_{BB} = 0.2$.

The images from the first row of each figure (Fig. 5 - Fig. 10) give visual confirmation of the previous conclusions derived from Table 1. and Fig. 4. Namely, we can see that with FastB algorithms we obtain clearer images (with less damage) than with Burst algorithms for both considered combinations of P_{GG} and P_{BB} and all different values of SNR. This confirms smaller values of BER (bit-error-probability) given in Table 1, and presented graphically in Fig. 4. Comparing the images before and after applying the filter (the corresponding images in the first and second row of each figure) we can conclude that the proposed filter enhances the quality of the images for all considered values of SNR.

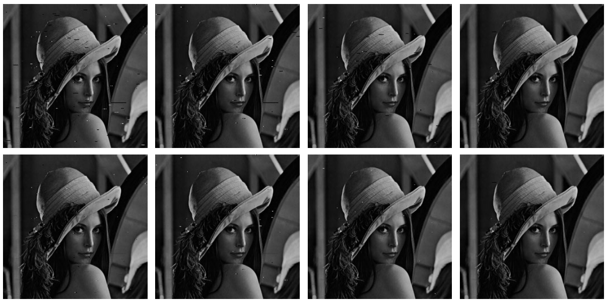


FIGURE 9. Images for $P_{GG} = 0.8$, $P_{BB} = 0.2$ and $SNR = -3$.

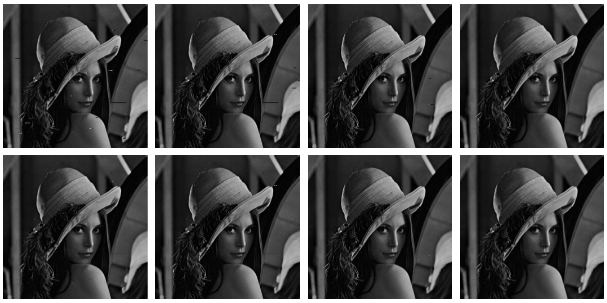


FIGURE 10. Images for $P_{GG} = 0.8$, $P_{BB} = 0.2$ and $SNR = -2$.

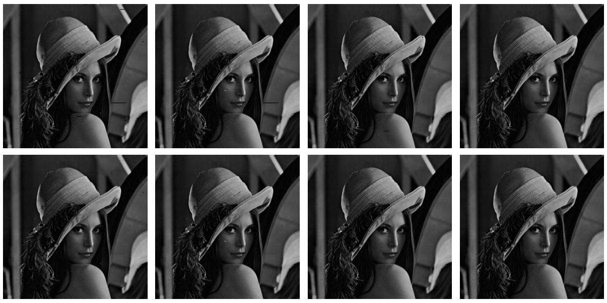


FIGURE 11. Images for $P_{GG} = 0.8$, $P_{BB} = 0.2$ and $SNR = -1$.

IV. CONCLUSION

In this paper, we consider the performances of cryptcodes based on quasigroups for image transmission through burst channels. We compare experimental results for codes obtained by Burst-Cut-Decoding, Burst-4-Sets-Cut-Decoding, FastB-Cut-Decoding and FastB-4-Sets-Cut-Decoding algorithms. The presented results confirm that Fast algorithms improve the performances of these cryptcodes for transmission over a burst channel. The best results are obtained by FastB-4-Sets-Cut-Decoding algorithm. For further improvement of the damaged parts of images, we apply a median filter on these parts in the decoded images. The considered codes and presented results can be useful for application in satellite digital video broadcasting (DVB-S) coding and encryption schemes.

REFERENCES

[1] T. Hwang and T. R. N. Rao, "Secret error-correcting codes (SECC)," *IETE J. Res.*, vol. 36, nos. 5–6, pp. 362–367, Sep. 1990, doi: 10.1080/03772063.1990.11436907.

[2] N. Zivic and C. Ruland, "Parallel joint channel coding and cryptography," *Int. J. Electr. Electron. Eng.*, vol. 4, no. 2, pp. 140–144, 2010.

[3] H. Cam, V. Ozduran, and O. N. Ucan, "A combined encryption and error correction scheme: AES-turbo," *Electrica*, vol. 9, no. 1, pp. 891–896, 2009.

[4] D. Abbasi-Moghadam and V. T. Vakili, "Enhanced secure error correction code schemes in time reversal UWB systems," *Wireless Pers. Commun.*, vol. 64, no. 2, pp. 403–423, May 2012, doi: 10.1007/s11277-010-0206-2.

[5] K. Viswanath and P. V. Pearlsy, "Cryptocoding system based on AES and concatenated coding scheme involving BCH and QC-LDPC," in *Proc. Int. Conf. Appl. Theor. Comput. Commun. Technol. (iCATccT)*, Davangere, India, Oct. 2015, pp. 189–194, doi: 10.1109/ICATCCT.2015.7456880.

[6] D. Gligoroski, S. Markovski, and L. Kocarev, "Error-correcting codes based on quasigroups," in *Proc. 16th Int. Conf. Comput. Commun. Netw.*, Honolulu, HI, USA, Aug. 2007, pp. 165–172, doi: 10.1109/ICCCN.2007.4317814.

[7] A. S. S. Popovska-Mitrovikj Markovski and V. Bakeva, "Increasing the decoding speed of random codes based on quasigroups," in *Proc. Web ICT Innov.*, S. Markovski, M. Gusev, Eds. Ohrid, Macedonia, 2012, pp. 93–102.

[8] A. Popovska-Mitrovikj, S. Markovski, and V. Bakeva, "4-sets-cut-decoding algorithms for random codes based on quasigroups," *AEU Int. J. Electron. Commun.*, vol. 69, no. 10, pp. 1417–1428, Oct. 2015.

[9] D. Mechkaroska, A. Popovska-Mitrovikj, and V. Bakeva, "New cryptcodes for burst channels," in *Algebraic Informatics CAI* (Lecture Notes in Computer Science), vol. 11545, M. Ciric, M. Droste, and J. É. Pin, Eds. Cham, Switzerland: Springer, 2019, pp. 202–212.

[10] A. V. V. Popovska-Mitrovikj Bakeva and D. Mechkaroska, "New decoding algorithm for cryptcodes based on quasigroups for transmission through a low noise channel," in *ICT Innovations*, vol. 778, D. Trajanov and V. Bakeva, Eds. Cham, Switzerland: Springer, 2017, pp. 196–204, doi: 10.1007/978-3-319-67597-8_19.

[11] A. V. V. Popovska-Mitrovikj Bakeva and D. Mechkaroska, "Fast decoding with cryptcodes for burst errors," in *ICT Innovations 2020, Machine Learning and Applications* (Communications in Computer and Information Science), vol. 1316, V. Dimitrova and I. Dimitrovski, Eds. Cham, Switzerland: Springer, 2020, pp. 162–173, doi: 10.1007/978-3-030-62098-1_14.

[12] D. Gligoroski, S. Markovski, and K. Lj, "Totally asynchronous stream ciphers + redundancy = Cryptcoding," in *Proc. Int. Conf. Secur. Manage.*, S. Aissi, H. R. Arabnia, Eds. Las Vegas, NV, USA, 2007, pp. 446–451.

[13] D. Mechkaroska, A. Popovska-Mitrovikj, and V. Bakeva, "A filter for images decoded using cryptcodes based on quasigroups," in *Proc. 14th Int. Conf. Informat. Inf. Technol.*, Mavrovo, Macedonia, 2017, pp. 52–56.

[14] J. Knag, W. Stark, and A. Hero, "Turbo codes for fading and burst channels," in *Proc. IEEE Theory Mini Conf.*, Feb. 1998, pp. 40–45.

[15] H. Labiod, "Performance of Reed Solomon error-correcting codes on fading channels," in *Proc. IEEE Int. Conf. Pers. Wireless Commun.*, Jaipur, India, Dec. 1999, pp. 259–263, doi: 10.1109/ICPWC.1999.759628.

ALEKSANDRA POPOVSKA-MITROVIKJ received the B.S. and M.S. degrees in computer science from the Faculty of Natural Sciences and Mathematics, Ss. Cyril and Methodius University in Skopje, in 2003 and 2009, respectively, and the Ph.D. degree in computer science from the Faculty of Computer Science and Engineering, Ss. Cyril and Methodius University in Skopje, in 2014.

Her Ph.D. dissertation is in the field of coding theory and cryptography. From 2002 and 2011, she has been working as a teaching/research assistant at the Faculty of Natural Sciences and Mathematics, Ss. Cyril and Methodius in Skopje. Since 2011, she has been working at the Faculty of Computer Science and Engineering, Ss. Cyril and Methodius in Skopje. Currently, she is an Associate Professor with the Faculty of Computer Science and Engineering. She has participated in several international and national scientific research projects, many international scientific conferences, workshops, and schools. She is the author or coauthor of one book, more than 40 scientific research papers published in journals and proceedings of international conferences, and two articles in a scientific journals with an impact factor. Her research interests include coding theory, cryptography, the application of ML and NLP in security, cryptography and coding, and blockchain technology.

Dr. Popovska-Mitrovikj received the Award from the Ss. Cyril and Methodius University in Skopje for graduation with the highest average grade at the Faculty of Natural Sciences and Mathematics in the academic year 2002/03.

DANIELA MECHKAROSKA received the B.S. degree from the Faculty of Science and Mathematics, Institute of Informatics, St. Cyril and Methodius University in Skopje, and the master's and Ph.D. degrees in coding and cryptography from the Faculty of Information Sciences and Computer Engineering, St. Cyril and Methodius University in Skopje.

After graduation, she worked in several companies in the field of informatics. In 1999, she founded the "Algorithm" Computer Center, which started with mathematics and computer classes and then turned into an educational center for foreign languages. Since 2020, she has been with the University of Information Sciences and Technologies "Saint Paul the Apostle," Ohrid, where she is currently an Assistant Professor. She is also a part-time Assistant Professor with International Balkan University, Skopje. She is the author or coauthor of 17 scientific papers and a participant in several domestic and foreign scientific conferences. Her research interests include coding theory, cryptography, cryptography and coding, blockchain technology, the IoT, and cloud computing services.

VERICA BAKEVA received the B.S. and M.S. degrees in mathematics and informatics and the Ph.D. degree in informatics from the Faculty of Natural Sciences and Mathematics, Ss. Cyril and Methodius University in Skopje, in 1992, 1998, and 2003, respectively.

Her Ph.D. thesis is in the field of applications of probabilistic models in queueing systems, cryptography, and coding theory. In the period between 1993 and 2003, she was a Teaching/Research Assistant with the Faculty of Natural Sciences and Mathematics, Ss. Cyril and Methodius University in Skopje. After defense of her Ph.D. thesis, in 2003, she was promoted to an Assistant Professor and in 2008, she was an Associate Professor with the Faculty of Natural Sciences and Mathematics. Since 2011, she has been with the Faculty of Computer Science and Engineering, Ss. Cyril and Methodius University in Skopje, where she was promoted to a Full Professor, in 2013. She has participated in several international and national scientific research projects, many international scientific conferences, workshops, and schools. She is the author of one university textbook, coauthor of five high school textbook, and coauthor of one monography. She is the author/coauthor of more than 60 scientific research papers published in journals and proceedings of international conferences. Her research interests include probabilistic models and their applications, especially in coding theory and cryptography.