

Short Survey of the Interleaver Technique in Communication System

Zeina Abdullah Humadi¹, Qusay Fadhel Al-Doori²

^{1,2}Control and Systems Engineering Department, University of Technology, Baghdad, Iraq

¹cse.19.11@grad.uotechnology.edu.iq, ²qusay.f.hasan@uotechnology.edu.iq

Abstract— Communication and computing systems have made it easier for the world to transfer data and information from the sender to the recipient at the lowest cost and most efficiency. The transmission process may cause data corruption or error for many reasons, including the environment, the large volume of transmitted data, heat, and noise. For these reasons, there is a need to correct and treat these errors. Individual errors can be easily corrected by hamming code, while burst errors cannot be corrected easily and need a hardware device called the interleaver used to correct the burst error. In this research, the different types of interleaver are studied and compared to find the best interleaver in order to increase the efficiency and accuracy of the systems. The issue is that interleaving takes a long time, which increases the turbo code's overall execution time. Our goal is to create an interleaver that is more sensitive and efficient than other varieties.

Index Terms— interleaving, Chaotic Interleaver, turbo codes, 2d baker map.

I. INTRODUCTION

Interleaver is a device that improves the efficiency of a system by rearranging data in one-to-one systems. Deinterleaver is the inverse of this operation, which restores the data to its original order. The interleaver is one of the most significant components in Turbo code, and it prevents burst errors, which helps to enhance overall Turbo code performance [1][2]. Interleaving allows for the elimination of connections between adjacent bits and the dispersion of error bits within blocks to reduce the impact of transmission-related factors like burst noise [3][4]. The ability to tolerate burst errors is improved by bit interleaving, which can break the inherent correlation in the input bit sequence through specific permutation rules and intelligently reorder the original bit sequence [5]. The design of Forward Error Correction (FEC) codes like the Turbo code [6][7] and LDPC code frequently employs bit interleaving. For turbo codes, an interleaver is used between the two component encoders. The interleaver is used to generate random input sequences [8]. In turbo encoders, interleaver is employed on the receiver side, and de-interleaver is used at the receiver side of turbo decoders. In order to enable forward-error correction (FEC) algorithms to recover the data, an interleaver spreads a series of bits over a longer period of time, ideally longer than the block size [9].

The aim of this research is to study and compare the types of interleaver to find the best in order to increase the efficiency and accuracy of the systems. The problem is the long time it takes to interleaving and that increases the execution time for the turbo code as a whole. Our goal is to get a quick responsive and more efficient interleaver compared to other types.

II. PRINCIPLE OF WORKING

Interleaving is a technique for spreading outbursts of errors by rearranging code symbols. An interleaving method converts a burst error in a memory channel into an arbitrary individual error without memory to correct errors [10][11]. During signal transmission, a burst of continuous errors appears on a segment of the code channel that is still active. The interleaved code is moderately discrete and randomized after the interleaving process, and these continuous errors can be subdivided into smaller errors.

III. INTERLEAVERS TYPES

Block and convolutional interleaves are the two types of traditional interleaves. The block has several types, as indicated in [12][13]. The following are the categories into which interleavers fall:

A. Block interleaver

A block interleaver moves coded symbols from the encoder and enters the rearranged symbols into the data modulator. To rearrange the blocks, the sequence is encoded in columns of an M-row by N-column (M X N) array. Following the filling of the array, these symbols are transmitted one row at a time into the modulator and broadcast over the channel. The symbols are entered one row at a time and removed one column at a time at the receiver. Before decoding, these code symbols are correctly deinterleaved at the receiver [13][14].

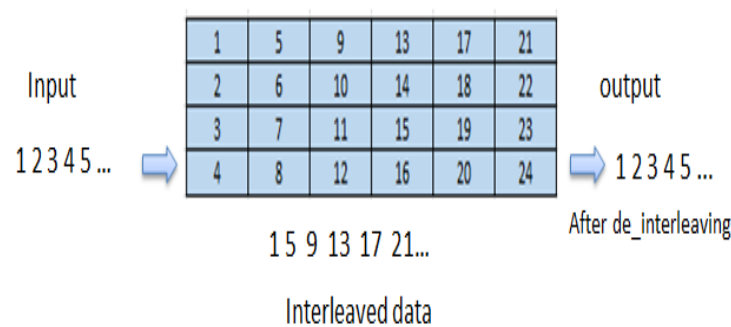


FIG. 1. BLOCK INTERLEAVER.

B. Matrix interleaver

It's a block type of interleaver that works row by row stuffing a matrix with input bits until the matrix is full, then column by column transferring the bits in the matrix to the output port [15][16].

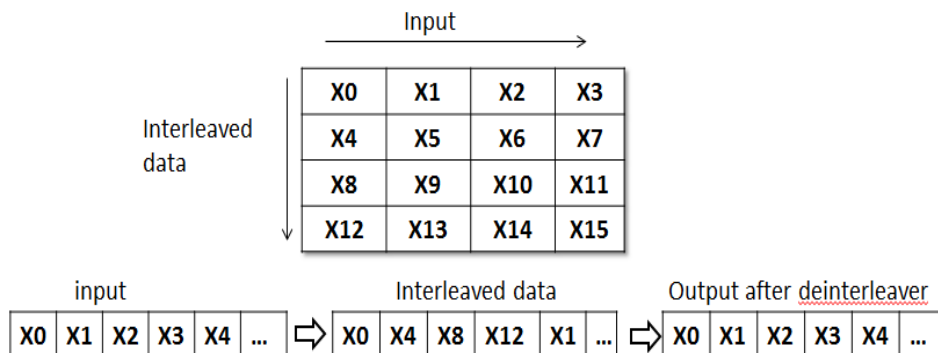


FIG. 2. MATRIX INTERLEAVER.

C. Random interleaver

The input vector is permuted at random based on an initial seed in this most basic type of interleaver. The error burst can be removed at the receiver and easily detected and corrected due to the randomization of elements. Because it randomly rearranges the input elements sequence based on the initial seed, each run time may produce a different output sequence. The process of restoring data is known as deinterleaving [11]. Because it uses lookup tables to implement interleaving, random interleaver has a major disadvantage. As a result, an algorithmic pseudo-random interleaver can be implemented, which eliminates the need for lookup tables and reduces hardware complexity for a simple implementation[17].

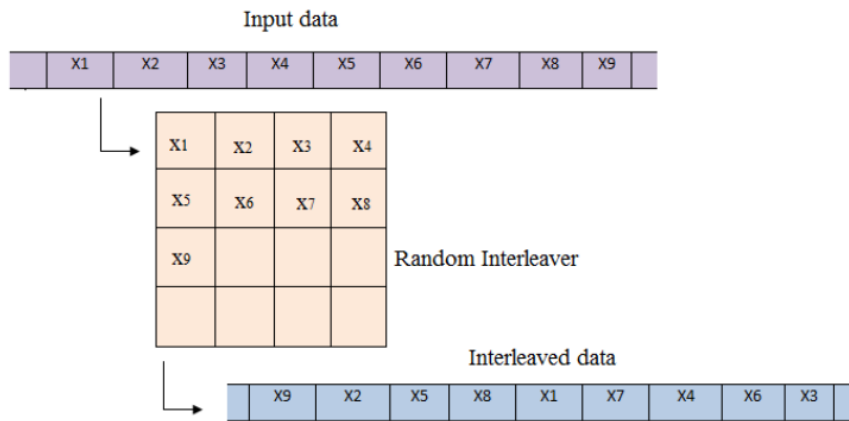


FIG. 3. RANDOM INTERLEAVER [13].

D. S Random Interleaver

In some situations, the address of data is randomly selected using a random interleaver. It's a hybrid of a block and a circular shifting interleaver. This interleaver is difficult to design because the complexity of achieving the condition of selecting the input data address requirement increases with the number of bits already tested[13].

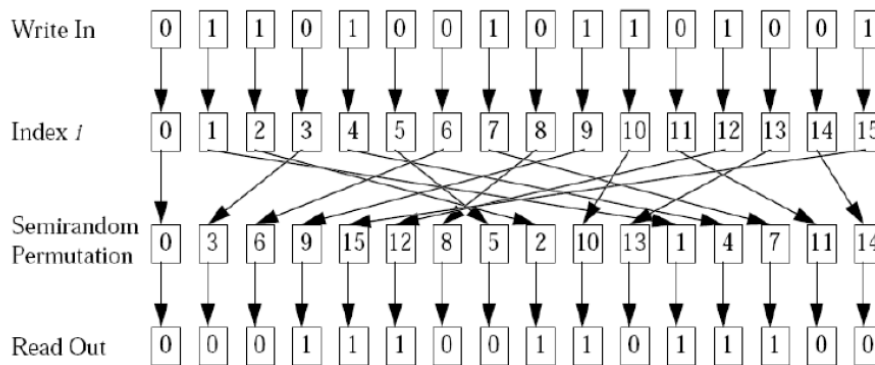


FIG. 4. S-RANDOM INTERLEAVER [17].

E. Odd-even interleaver

This interleaver should have an odd number of columns and rows. Because the symbols are left un-interleaved and encoded, only the odd-location coded symbols are loaded at first. Only the symbols with even-location codes are loaded after the symbols are randomized and encoded. This interleaver is designed to achieve a half-code rate by puncturing two non-systematic codes [18].

Odd-even interleaver output														
Encoder output without interleaving														
X ₁	X ₂	X ₃	X ₄	X ₅	X ₆	X ₇	X ₈	X ₉	X ₁₀	X ₁₁	X ₁₂	X ₁₃	X ₁₄	X ₁₅
Y ₁	-	Y ₃	-	Y ₅	-	Y ₇	-	Y ₉	-	Y ₁₁	-	Y ₁₃	-	Y ₁₅
Encoder output with row-column interleaving														
X ₁	X ₆	X ₁₁	X ₂	X ₇	X ₁₂	X ₃	X ₈	X ₁₃	X ₄	X ₉	X ₁₄	X ₅	X ₁₀	X ₁₅
-	Z ₆	-	Z ₂	-	Z ₁₂	-	Z ₈	-	Z ₄	-	Z ₁₄	-	Z ₁₀	-
Final Odd-even interleaver output														
Y ₁	Z ₆	Y ₃	Z ₂	Y ₅	Z ₁₂	Y ₇	Z ₈	Y ₉	Z ₄	Y ₁₁	Z ₁₄	Y ₁₃	Z ₁₀	Y ₁₅

FIG. 5. ODD-EVEN INTERLEAVER OPERATION [15].

F. Algebraic interleaver

The Algebraic Interleaver uses an algebraically defined permutation to rearrange the symbols in its input sequence. As a result, the complexity and latency effect of algebraic interleavers are low, and it can be implemented using basic blocks. The algebraic interleaver mathematical expression is

$$\Pi(k) = k \text{MOD} N \tag{1}$$

where N is the interleaver length, $\Pi(k)$ is a number between $[0, N - 1]$, MODN modulo N arithmetic [19][20].

G. Helical interleaver

It reads data diagonally and writes data row by row. Fig. 6 illustrates this. The number of columns in a helical array can be expressed by M, therefore the array has M columns and limitless rows. If the group size is N, the block will accept an MN-bit input at each time step and divide it into N-bit sets. Because of the modulo M reduction and the knowledge that the set's initial bit is in row 1, (j-1). b, the situation is helical, where b is the helical array's step size [21][22].

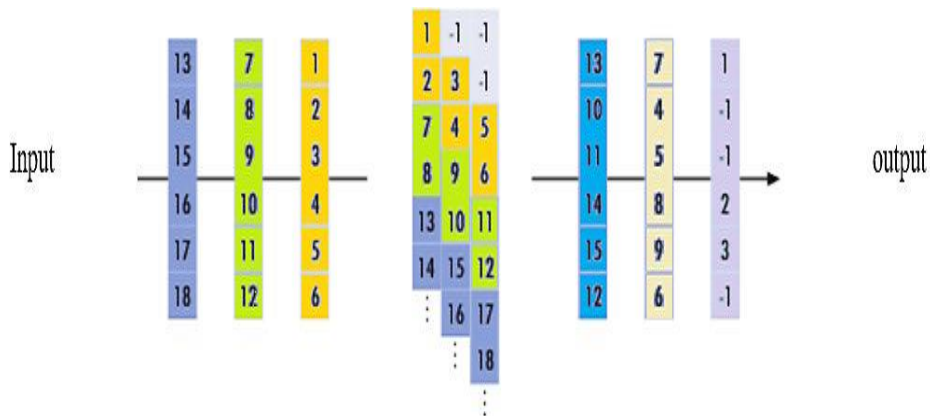


FIG. 6. HELICAL INTERLEAVER[15].

Number of columns = 3, group size = 2

H. Matrix helical interleaver

It reduces the number of errors performed during a burst while also improving the ability to correct them. To generate helical interleaver, the interleaver indices are loaded on a regular basis in this type. Helical interleaver elements are first sorted by column and row, then loaded diagonally, as seen in Fig. 10. The formula $L = kr * kc$, can be used to calculate the length of an interleaver, where kr and kc

DOI: <https://doi.org/10.33103/uot.ijccce.23.2.3>

are the row and column sizes, respectively. The array step-size argument is the amount by which the row index rises when the column index grows by one [15].

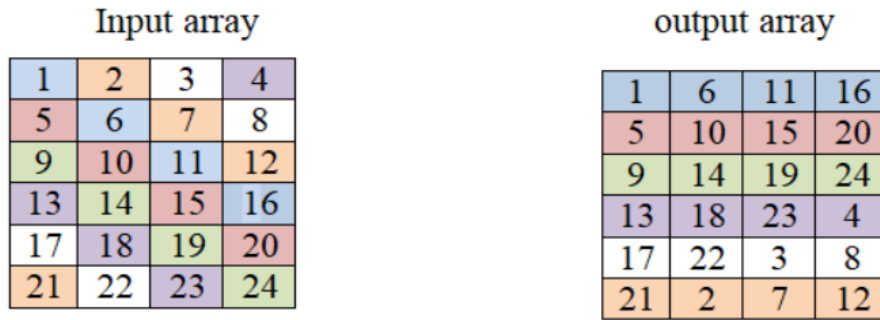


FIG. 7. OPERATION OF MATRIX HELICAL INTERLEAVER.

I. Convolutional Interleaver

A convolutional interleaver [23] in Fig. 11 is a type of interleaver that is made up of many shift registers. Each of the shift registers has a fixed delay that is multiple of a fixed integer in positive integers. The next shift register is fed every time new data is fed into the interleaver's input, the previous data in that register is then included in the interleaver output. A convolutional interleaver [24] has memory, which means it can work with both current and previous symbols.

A standard convolutional interleaver has delays that are nonnegative multiples of a fixed integer (although a general multiplexed interleaver allows free delay values).

These interleavers when compared to block interleaves use less memory. When compared to the block interleaver, the convolutional interleaver requires half the memory.

A general representation of a convolutional interleaver's permutation is as follows:

$$i = \pi(k \bmod N) + \alpha(k \bmod N)N + \text{floor}\left(\frac{k}{N}\right)N \quad (2)$$

Where π is basic permutation with a finite number of possibilities of length N, The index for the original data is k, the shift vector with N components is, and the new position of the k symbol is i. mod denotes the operation modulus., and the greatest integer below the parameter is marked by floor.

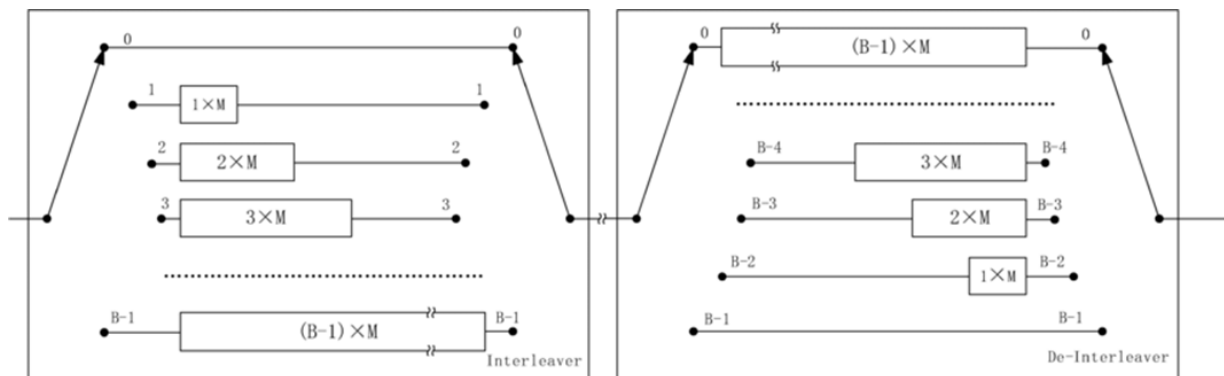


FIG. 8. CONVOLUTIONAL INTERLEAVER.

DOI: <https://doi.org/10.33103/uot.ijccce.23.2.3>

J. Zigzag Interleaver

Zigzag interleaving is appealing for maximum data rate requirements because of its reduced encoding and decoding complexity and perfect performance [25]. Fig. 7 (a) and 7 (b) show how zigzag patterns can be used to organize information symbols (b). Fig. 7 (c) depicts the zigzag interleaving operation when a 2-D error burst appears as a shaded area. The burst error becomes random, as shown in Fig. 6 (d). As a result, using the single error correction procedure to partially correct this burst error is possible. As a result, the Zigzag interleaving approach is capable of effectively removing 2D bursts of mistakes [26].

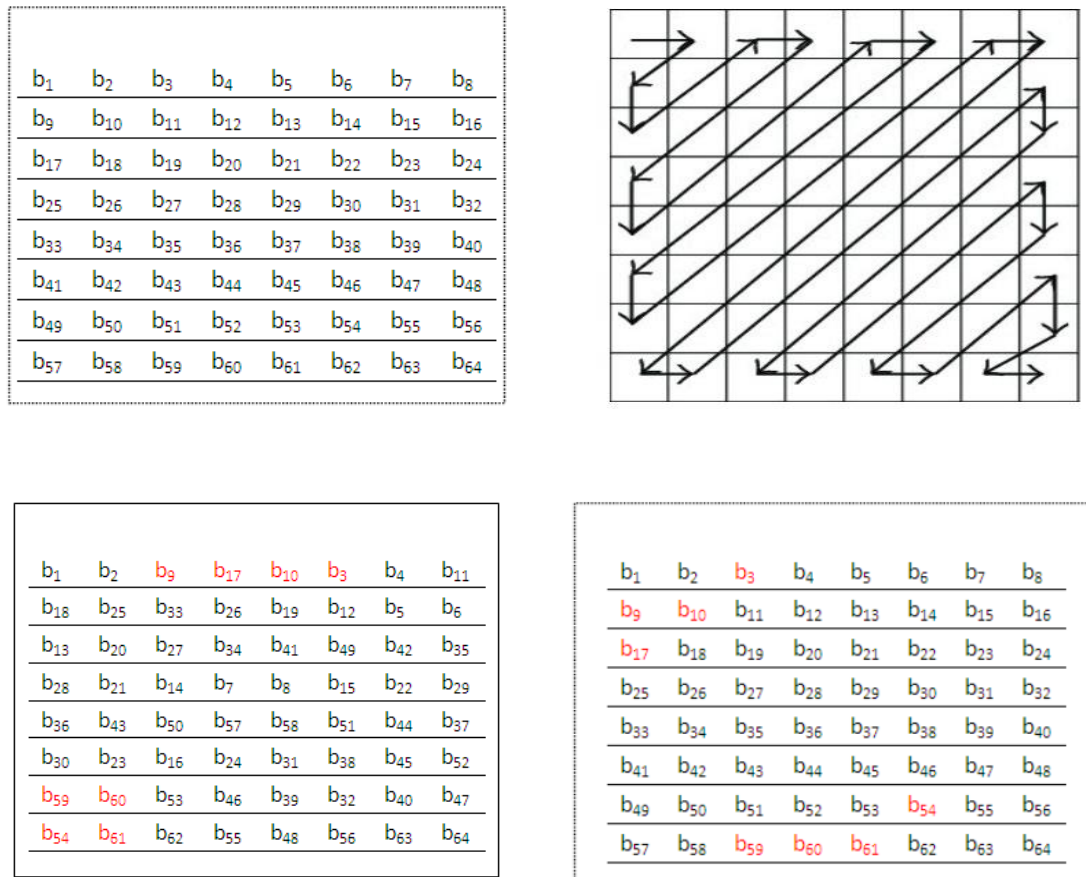


FIG. 9. ZIG-ZAG INTERLEAVING OF 8x8 MATRIX; (A) THE 8x8 MATRIX, (B) ZIGZAG RANDOMIZATION PROCEDURE, (C) ZIGZAG INTERLEAVING OPERATION, (D) ERROR BURSTS REMOVED AFTER DEINTERLEAVING[10].

K. Chaotic Interleaver

Baker map [27] is used to create the chaotic interleaver. There are other types of maps for chaotic, for example, henon chaotic map, and logistic map [28], but we focus in this paper on 2-D baker map. The (generalized) baker's map is a basic two-dimensional map that may be used to illustrate many of the ideas for describing chaos. The block type is the most basic and well-known interleaving technique. However, with 2-D error bursts, this type is ineffective. As a result, developed interleavers must be used for this function. As a result, the discrete version of the two-dimensional chaotic Baker map is a great choice. Before being randomized with the chaotic Baker map, the elements can be rearranged into a square matrix. To randomize the bits in a 2-D geometry, the Baker map is employed [29][30].

DOI: <https://doi.org/10.33103/uot.ijccce.23.2.3>

1_D burst error

B1	B2	B3	B4	B5	B6	B7	B8
B9	B10	B11	B12	B13	B14	B15	B16
B17	B18	B19	B20	B21	B22	B23	B24
B25	B26	B27	B28	B29	B30	B31	B32
B33	B34	B35	B36	B37	B38	B39	B40
B41	B42	B43	B44	B45	B46	B47	B48
B49	B50	B51	B52	B53	B54	B55	B56
B57	B58	B59	B60	B61	B62	B63	B64

2_D burst error

FIG. 10. (A) 8*8 MATRIX BURST ERROR WITHOUT INTERLEAVING.

B1	B9	B17	B25	B33	B41	B49	B57
B2	B10	B18	B26	B34	B42	B50	B58
B3	B11	B19	B27	B35	B43	B51	B59
B4	B12	B20	B28	B36	B44	B52	B60
B5	B13	B21	B29	B37	B45	B53	B61
B6	B14	B22	B30	B38	B46	B54	B62
B7	B15	B23	B31	B39	B47	B55	B63
B8	B16	B24	B32	B40	B48	B56	B64

(B) AFTER BLOCK INTERLEAVING 8*8 MATRIX.

B31	B23	B15	B7	B32	B24	B16	B8
B63	B55	B47	B39	B64	B56	B48	B40
B29	B21	B13	B5	B30	B22	B14	B6
B61	B53	B45	B37	B62	B54	B46	B38
B27	B19	B11	B3	B28	Z	B12	B4
B59	B51	B43	B35	B60	B52	B44	B36
B25	B17	B9	B1	B26	B18	B10	B2
B57	B49	B41	B33	B58	B50	B42	B34

(C) CHAOTIC INTERLEAVING 8*8MATRIX.

Through the geometric figure, we note the previous one shows us that in the chaotic interleaving it is better to treat 1_d and 2_d burst errors than the block interleaving

Chaotic interleaver can be defined as a method of rearranging the elements of a square matrix chaotically. Let $B(n_1, n_2, \dots, n_k)$ represent the discretized map in its mathematical formula. The vector $[n_1, n_2, \dots, n_k]$ S key represents the sector key. The secret key is chosen so that each integer n_i divides N data items, using N as the number of data items in one row. according to the following condition:

$$(r,s) = Nn_i r - Ni + smod Nn_i, mod Nn_i + Ni \tag{3}$$

The algorithm for chaotic interleaving is as described in the following:

- The $M \times M$ matrix is divided into I several orthogonally rectangular shapes of height M and width m_k , where $m_1 + m_2 + \dots + m_i = M$.
- The horizontal orientations of these perpendicular rectangles will be extended and organized perpendicularly to produce a $m_k \times M$ horizontal rectangle.
- The base of these rectangle forms is on the left, and the higher section is on the right, as shown in Fig. 5 (a).

DOI: <https://doi.org/10.33103/uot.ijccce.23.2.3>

- The dimensions $\frac{M}{m_k} \times m_k$ and the exact number of points M are divided into n_i boxes for each perpendicular rectangle $m_k \times M$.

An example is provided to help you understand these stages. As described in Fig. (2-12) below at (a), (b), and (c), a basic square matrix of 8x8 equals 64 items (c). The designer must select a key that satisfies the chaotic interleaver condition, which divides the square matrix into N-element rectangles. In this case, N=8. As a result, the selected key = [2, 4, 2]. Heavy strong lines define the rectangular borders. A chaotic interleaver with a 16*16 square matrix equals 256 elements. Key = [2, 2, 4, 4, 2, 2], with each inside square matrix being a rectangle with N=16 elements, is the suggested key that satisfies chaotic interleaver requirements.

The input frame length of 1024 bits is one of the simulation parameters. This necessitates the creation of a square matrix with a 32 x 32 dimension. The data input is arranged in a matrix. The next step is to make key division rectangles in the square matrix, with a total of N = 32 elements. [2, 2, 2, 2, 4, 4, 2, 2, 2, 2] is the key. By reading the new sequence of input data row by row, the chaotic interleaved frame of length 1024 bits is created.

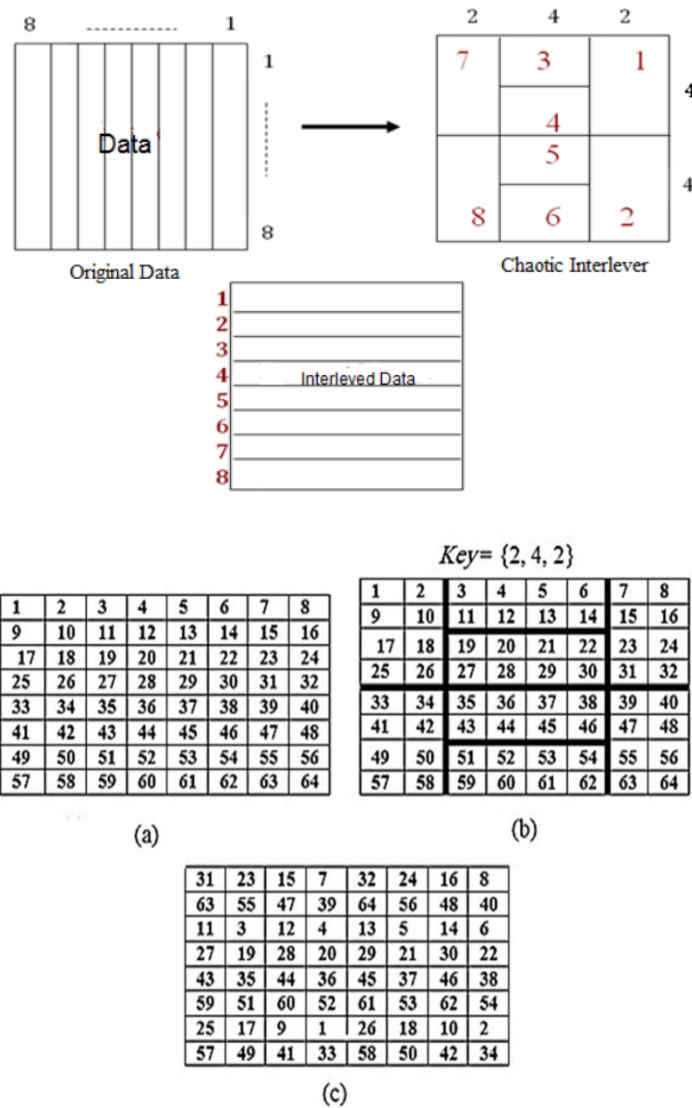


FIG. 11. CHAOTIC INTERLEAVER ON 8x8 SQUARE MATRICES, (A) ORIGINAL INPUT SEQUENCE, (B) KEY SEGMENTS, AND (C) CHAOTIC INTERLEAVING.

DOI: <https://doi.org/10.33103/uot.ijccce.23.2.3>

Many important characteristics of chaotic processes include instability of the initial state and system parameter, ergodicity, and confusion, among others. Because of these characteristics, chaotic systems are an excellent choice for creating cryptosystems. The instability of the starting status/system parameter and mixing features, respectively, are equivalent to the confusion and diffusion features of a perfect cryptosystem [3]. Chaotic systems offer a wide range of applications [31], including biology and communication [32][33].

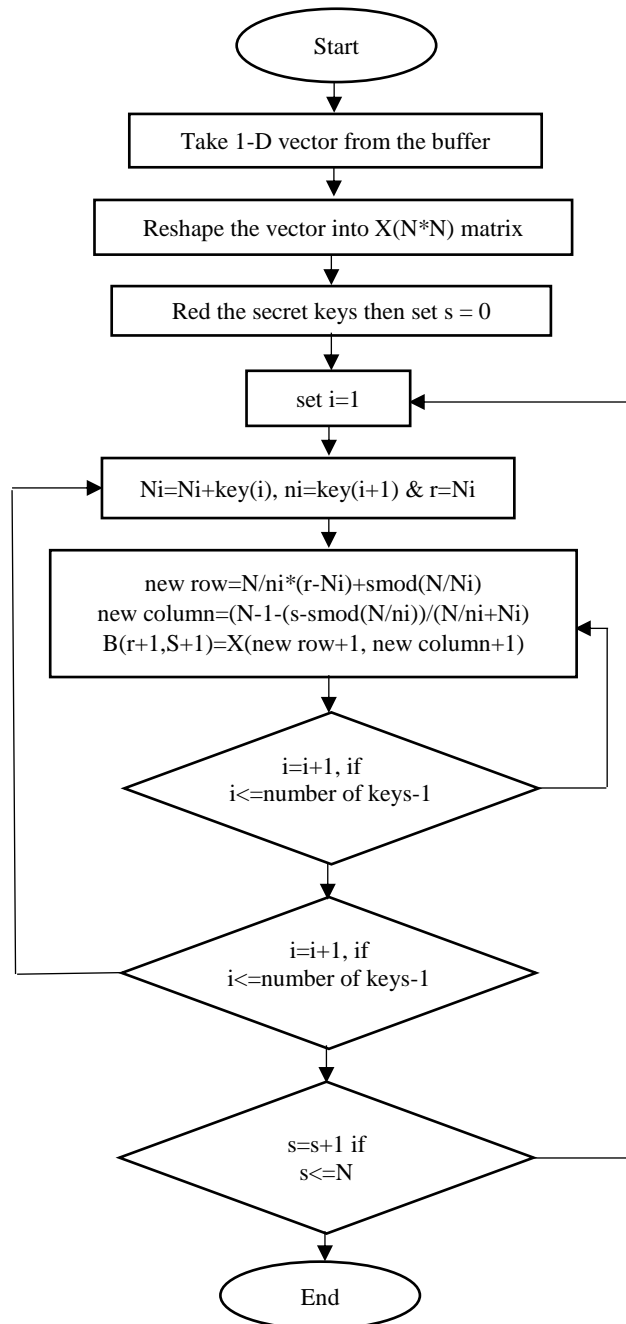


FIG. 12. THE FLOWCHART OF THE CHAOTIC BAKER MAP ALGORITHM.

DOI: <https://doi.org/10.33103/uot.ijccce.23.2.3>**Algorithm: chaotic baker map**

- 1- Take 1D vector from the buffer
- 2- Reshape the input matrix from 1-D to 2-D
- 3- Define a new matrix equal to zero whose size is the same as the size of the input
- 4- Put key=[2 , 4 ,2] when the matrix input =8 ,
Key = [2, 2, 4, 4, 2, 2] when the input of matrix = 16 and
Key = [2, 2, 2, 2, 4, 4, 4, 4, 2, 2, 2, 2] when the input of matrix =32
- 5- for s=0:row-1
 Ni=0;
- 6- for i=1:key-1
 Ni=Ni+Sk (i);
 ni= Sk(i+1);
- 7- for r=Ni:Ni+ni-1
- 8- q=N/ni;
- 9- New row = q*(r-Ni)+mod(s,q);
New column = (N-1)-((s-mod(s,q))/q+Ni);
 Interleaver(r+1, s+1) = input (nrow+1, ncol+1);
- 10- end

TABLE I. METHODS AND TYPES OF INTERLEAVER USED IN RESEARCH

Item	Method	Interleaver type	Optimization
Wang and Li (2007)	In wideband CDMA (W-CDMA) systems, this brief describes a low-complexity hardware interleaver solution for turbo code. To reduce computing complexity and latency, algorithmic transformations are often used. New VLSI architectures are being created[34].	Random	The results of the hardware implementation demonstrate that a complete turbo interleave pattern generation unit uses just 4 k gates, which is an order of magnitude less than traditional designs.
Khater , Khairty and Habib (2009)	In this paper, the author claims that interleaving is important for improving the bit error rate performance of the forward error correction (FEC) mechanism. The process of rearranging code symbols in order to distribute a series of errors into random errors that can be corrected using the FEC algorithm is known as interleaving.[35].	Block	The area and delay efficiency in this new architecture
Vosoug hi et al (2013)	They propose a new algorithm and architecture for the highly scalable UMTS/HSPA+ standard's on-the-fly generation of parallel interleaved addresses[36].	Random	Original addresses can be used to identify interleaved addresses. Our method also includes a hardware architecture that is both efficient and scalable.
Omeira , Hamad and Elbayou my (2015)	This paper introduces a new collision-free S-random interleaver design. The proposed interleaver combines the high performance of code-matched interleavers with the property of collision-free interleaving. [37].	S-random	The presented interleaver has a better error performance than the non-codematched collision-free random interleaver.

DOI: <https://doi.org/10.33103/uot.ijccce.23.2.3>

J.sharma , Mishra and sharma (2017)	The types of interleavers are examined in this paper. In addition, using the Maximum A-Posteriori (MAP) algorithm, a comparative analysis of the main types of interleaver is performed for better bit error rate (BER) performance in linear turbo equalizer[13].	Various types	Block interleaver achieves better performance than other interleavers for high SNR and thus low bit error (BER).
Patel (2020)	This paper compares various interleavers with coded and uncoded IDMA to improve the performance of an IDMA system using an unequal power algorithm[38].	Various types	In terms of memory and complexity, the simulation results show that the IDMA system performs better with invert tree based interleaver (ITBI) than other interleavers.
Sahnoun e and Berkani (2021)	In this paper they describe a new method for designing a deterministic interleaver with random-like behavior based on a two-dimensional chaotic map, known as the lozi map. [39].	Chaotic and random	The statistical properties and performance of such an interleaver were studied and compared to random interleaver. The use of a chaotic interleaver reduces latency and implementation complexity while also improving the communication system's reliability and security.

Interleaver type	Features
Convolutional	In fact, the achievement of convolutional interleaver is close to the performance of block interleaver, but there are some characteristics that distinguish them from each other. For example, Convolutional interleaver needs half of the delay and memory compared to the block interleaver.
Block	The block interleaver also minimizes device utilization and power usage. As mentioned in this research. They discovered that the block interleaver performed better than the other varieties. As a result, a block interleaver is considered to be effective in enhancing connection performance.
Helical	Because it writes data row by row and reads data diagonally, there are empty sites where -1 is used in this type.
Matrix helical	In performance and operation, it is close to the helical type, but the type is better than the helical interleaver because it enters the data in the form of an array, meaning that there are no empty places for this reason, the delay in this type is less than the helical.
Matrix	The numbers n and m identify a matrix interleaver, which is also known as a (n, m). Both n and m have an impact on the ability to disperse burst errors.
Odd-even	This interleaver should have an odd number of columns and rows.
S Random	This interleaver is difficult to design because the complexity of reaching the condition of selecting the input data address requirement increases with the number of bits already tested.
Random	Because it is dependent on the initial seed and characteristic randomization, the error burst can be removed at the receiver and easily detected and corrected, this type is the best for the types mentioned earlier in this comparison. But when compared to the chaotic interleaver it has the disadvantage that it needs a lookup and this makes the system more complex

DOI: <https://doi.org/10.33103/uot.ijccce.23.2.3>

Chaotic	When designing a chaotic interleaver, there is no need to transmit the entire interleaver look-up table, which reduces memory usage and latency. Chaotic interleaver has lower latency, lower implementation complexity, and enhances the security of the encoded data when compared to random interleaver.
---------	---

IV. CONCLUSIONS

In this paper, we studied the types of interleaver and compared them and found that the two best types are the random interleaver and the chaotic interleaver. And the chaotic interleaver is superior to the random interleaver, because the random depends on the look-up table and this makes the system less fast and more complex. While there is no need to transmit the entire interleaver look-up table when designing a chaotic interleaver, this minimizes memory usage and latency. Chaotic interleaver has lower latency, lower implementation complexity, and enhances the security of the encoded data when compared to random interleaver.

REFERENCES

- [1] C. Berrou, "The ten-year-old turbo codes are entering into service," *IEEE Commun. Mag.*, vol. 41, no. 8, pp. 110–116, 2003.
- [2] M. A. Fleah and Q. F. Al-Doori, "Design and Implementation of Turbo encoder/decoder using FPGA," *1st Int. Sci. Conf. Comput. Appl. Sci. CAS 2019*, pp. 46–51, 2019, doi: 10.1109/CAS47993.2019.9075589.
- [3] K. Huo, Z. Hu, and D. Liu, "Design and Implementation of Shared Memory for Turbo and LDPC Code Interleaver," *Wirel. Commun. Mob. Comput.*, vol. 2022, 2022.
- [4] G. A. Jaquenod and G. G. Acosta, "A simple block interleaving algorithm using reduced memory and address generator resources," in *2020 Argentine Conference on Electronics (CAE)*, 2020, pp. 53–56.
- [5] Y. Ould-Cheikh-Mouhamedou, "A simple and efficient method for lowering the error floors of turbo codes that use structured interleavers," *IEEE Commun. Lett.*, vol. 16, no. 3, pp. 392–395, 2011.
- [6] M. Sreedevi, B. Yamuna, and P. Salija, "Design and Implementation of Interleaver in GNU Radio for short block length Turbo codes," in *2019 9th International Conference on Advances in Computing and Communication (ICACC)*, 2019, pp. 17–21.
- [7] R. Li, J. Tian, X. Bian, and M. Li, "CRCSI: A Generic Block Interleaver for the Next Generation Terrestrial Broadcast Systems," *Appl. Sci.*, vol. 12, no. 4, p. 2025, 2022.
- [8] M. A. Fleah and Q. F. Al-Doori, "State Space Parallelization Method for a 16-Bit Turbo Encoder," *Eng. Technol. J.*, vol. 37, no. 12A, pp. 553–557, Dec. 2019, doi: 10.30684/etj.37.12A.9.
- [9] M. Kubiak and E. Grayver, "High Throughput Multi-Threaded Software Defined Convolutional Interleaver," in *2021 IEEE Aerospace Conference (50100)*, 2021, pp. 1–14.
- [10] V. Chaturvedi, V. K. Gupta, and D. I. T. Dehradun, "Performance Analysis for Different Interleavers in Various Modulation Schemes with OFDM over an AWGN Channel," *IOSR J. Eng.*, vol. 2, no. 4, pp. 760–767, 2012.
- [11] B. Das, M. P. Sarma, K. K. Sarma, and N. Mastorakis, "Design of a few interleaver techniques used with gold codes in faded wireless channels," in *2015 2nd International Conference on Signal Processing and Integrated Networks (SPIN)*, 2015, pp. 237–241.
- [12] S. Sharma, P. C. Sau, and A. Shukla, "Performance survey of IDMA with different interleavers," in *2014 International Conference on Signal Processing and Integrated Networks (SPIN)*, 2014, pp. 344–348.
- [13] J. Sharma, S. Mishra, and N. Sharma, "An analysis of interleaver types for better BER performance in linear turbo equalizer," in *2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIECS)*, 2017, pp. 1–5.
- [14] "Difference between Block Interleaver Convolutional Interleaver." <https://www.rfwireless-world.com/Terminology/Difference-between-Block-Interleaver-and-Convolutional-Interleaver.html> (accessed Sep. 17, 2021).
- [15] M. J. M. Ameen and H. J. Kadhim, "The efficient interleaving of digital-video-broadcasting-satellite 2nd generations system," *Telkomnika*, vol. 18, no. 5, pp. 2362–2370, 2020.
- [16] J. Jeong, Y. Jeon, and D. Yoon, "Blind estimation of block Interleaver parameters using statistical characteristics," *Futur. Gener. Inf. Technol. 2016*, 2016.
- [17] L. Hanzo, T. H. Liew, and B. L. Yeap, *Turbo coding, turbo equalisation and space-time coding*. John Wiley & Sons, 2002.
- [18] K. S. Arkoudogiannis and C. E. Dimakis, "Performance analysis of the odd–even uniform interleaver for turbo codes," *IET Commun.*, vol. 13, no. 16, pp. 2469–2477, 2019.

DOI: <https://doi.org/10.33103/uot.ijccce.23.2.3>

- [19] O. Y. Takeshita and D. J. Costello, "New classes of algebraic interleavers for turbo-codes," in *Proceedings. 1998 IEEE International Symposium on Information Theory (Cat. No. 98CH36252)*, 1998, p. 419.
- [20] D. Hao and P. A. Hoeher, "Helical interleaver set design for interleave-division multiplexing and related techniques," *IEEE Commun. Lett.*, vol. 12, no. 11, pp. 843–845, 2008.
- [21] S. Ramabadran, A. S. Madhukumar, N. W. Teck, and C. M. S. See, "Parameter estimation of convolutional and helical interleavers in a noisy environment," *IEEE Access*, vol. 5, pp. 6151–6167, 2017.
- [22] B. Sklar, *Digital communications: fundamentals and applications*. 2001.
- [23] S. Sreelekha and G. Suchitra, "BER Performance Analysis of Block Interleaver and Convolutional Interleaver in QAM modulation using LabVIEW," *Hindusthan J. Inf. Commun. Mod. Comput.*, vol. 1, no. 1, pp. 19–25, 2020.
- [24] C. R. Sánchez-Ortiz, R. Parra-Michel, and M. E. Guzmán-Rentería, "Design and implementation of a multi-standard interleaver for 802.11 a, 802.11 n, 802.16 e & DVB standards," in *2008 International Conference on Reconfigurable Computing and FPGAs*, 2008, pp. 379–384.
- [25] G. Bauch and K. Kusume, "Interleaving strategies for concatenated zigzag codes," in *2007 16th IST Mobile and Wireless Communications Summit*, 2007, pp. 1–5.
- [26] B. Das, M. P. Sarma, and K. K. Sarma, "Different Aspects of Interleaving Techniques in Wireless Communication," in *Intelligent Applications for Heterogeneous System Modeling and Design*, IGI Global, 2015, pp. 335–374.
- [27] M. A. M. M. El-Bendary and A. A. El-Azm, "An efficient chaotic interleaver for image transmission over IEEE 802.15. 4 Zigbee network," *J. Telecommun. Inf. Technol.*, pp. 67–73, 2011.
- [28] A. M. Raheema, S. B. Sadkhan, and S. M. A. Sattar, "Design and Implementation of Speech Encryption Based on Hybrid Chaotic Maps," *2018 Int. Conf. Eng. Technol. their Appl.*, no. 1, pp. 112–117, 2018, doi: 10.1109/IICETA.2018.8458098.
- [29] W. Hadi, H. Hussein, "Image Security Over Wireless Sensor Network," *Mansour Journal*, Vol. 28, No. 1, 2017, doi: 10.36541/0231-000-028-007.
- [30] A. Ali Jassim and W. A.H. Hadi, "Performance Comparison of Proposed Interleaver With Different Types for Parallel Turbo Code," *J. Eng. Sustain. Dev.*, vol. 2018, no. 04, pp. 143–156, 2018, doi: 10.31272/jeads.2018.4.11.
- [31] F. Beritelli, E. Di Cola, L. Fortuna, and F. Italia, "Multilayer chaotic encryption for secure communications in packet switching networks," in *WCC 2000-ICCT 2000. 2000 International Conference on Communication Technology Proceedings (Cat. No. 00EX420)*, 2000, vol. 2, pp. 1575–1582.
- [32] F. Nazarimehr, S. Jafari, S. M. R. Hashemi Golpayegani, M. Perc, and J. C. Sprott, "Predicting tipping points of dynamical systems during a period-doubling route to chaos," *Chaos An Interdiscip. J. Nonlinear Sci.*, vol. 28, no. 7, p. 73102, 2018.
- [33] A. K. Farhan, N. M. G. Al-Saidi, A. T. Maolood, F. Nazarimehr, and I. Hussain, "Entropy analysis and image encryption application based on a new chaotic system crossing a cylinder," *Entropy*, vol. 21, no. 10, pp. 1–14, 2019, doi: 10.3390/e21100958.
- [34] Z. Wang and Q. Li, "Very low-complexity hardware interleaver for turbo decoding," *IEEE Trans. Circuits Syst. II Express Briefs*, vol. 54, no. 7, pp. 636–640, 2007.
- [35] A. A. Khater, M. M. Khairy, and S.-D. Habib, "Efficient FPGA implementation for the IEEE 802.16 e interleaver," in *2009 International Conference on Microelectronics-ICM*, 2009, pp. 181–184.
- [36] A. Vosoughi, G. Wang, H. Shen, J. R. Cavallaro, and Y. Guo, "Highly scalable on-the-fly interleaved address generation for UMTS/HSPA+ parallel turbo decoder," in *2013 IEEE 24th International Conference on Application-Specific Systems, Architectures and Processors*, 2013, pp. 356–362.
- [37] M. S. Omeira, G. M. Hamad, and A. D. Elbayoumy, "A code-matched collision-free S-random interleaver for turbo codes," in *2015 IEEE Seventh International Conference on Intelligent Computing and Information Systems (ICICIS)*, 2015, pp. 398–404.
- [38] A. Patel, "Interleave-Division Multiple Access Systems with Invert Tree Based Interleavers with Unequal Power Sharing Algorithm," in *2020 International Conference on Inventive Computation Technologies (ICICT)*, 2020, pp. 700–704.
- [39] A. Sahnoune and D. Berkani, "On the performance of chaotic interleaver for turbo codes," *SN Appl. Sci.*, vol. 3, no. 1, 2021, doi: 10.1007/s42452-021-04147-w.