

Huffman Code via. Fuzzy Generators

Ameer A.J. Al-Swidi, Enas Hamood Al-Saadi and Raid Abd Alreda Shekan
 Department of Mathematics, College of Education for Pure Science,
 University of Babylon, Hillah, Iraq

Abstract: In this study take the plaintext (characters) and applying Huffman code to convert the plaintext (characters) to plaintext (binary system) in Huffman code every character take code different from other character in addition more flexibility to deal us with key-generations (to produce key stream) like Geffe generator, threshold generator, Feedback with Carry Shift Register generator and FCSR combining generator with XOR's operation and second by fuzzy operation to encryption, the last evaluation measure the encryption by using the SNR and PSNR.

Key words: SNR, PSNR, LFSR, FCSR, stream, Huffman code

INTRODUCTION

A feedback shift register is commonly use method into cryptography and encoding state that has shortcut (FBSR), its operates by two phase, first phase, shift register and second phase, feedback function a sequence of shift register used in both coding theory, cryptography. For shift register the length is figured in n bits long and it is known as n-bits shift register (Al-Swidi and Al-Talkany 2014; Arnault *et al.*, 2008; Lee, 1990; El Abbadi and Al-Saadi, 2016; Goresky and Klapper, 2002).

By Huffman (1952) unable to prove any codes were the most efficient where his idea to sorted the data as binary tree with reduces the frequency.

MATERIALS AND METHODS

Preliminaries

Huffman code: Huffman codes are widely used for compressing data saving of 20-90% are typical. Depending on the data being compressed as the characteristics. Huffman code uses algorithm as table of the frequencies occurrence of the characters to build up an optimal way of representing each character as a binary string.

Is a type of optimal prefix code which are uses for lossless data redundancy the output algorithm of Huffman's can be viewed for encoding a source symbol as a variable-length code table from the frequency of occurrence or estimated probability the algorithm derives this table and uses a specific method for each possible value of the each symbol, resulting in a prefix code

Table 1: Frequency

Characters	Frequency	Code
R	3	00
I	2	01
A	1	100
Q	1	1010
V	1	1011
E	1	110
Space (-)	1	111

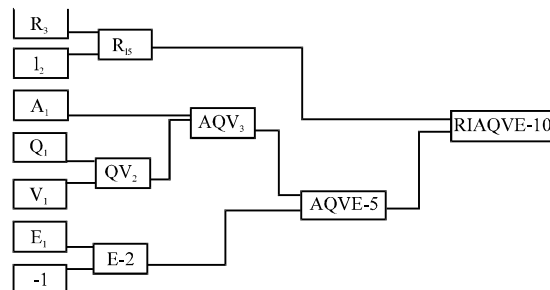


Fig. 1: tree of Huffman code

(prefix-free code) that is representing some symbol as bit string is never a prefix representing any other symbol as bit string (Huffman, 1952) (Table 1). For example, if the plaintext is (IRAQ RIVER) then the Huffman code will be is (Fig. 1) tree code.

Fuzzy coding: Fuzzy logic in 1965 was introduced by Zadeh in his study "fuzzy sets", zadeh continued to develop fuzzy logic with others at that time in this study take the disjoint sum of fuzzy logic ($a \Delta b = \min \{ \max (a, 1-b), \max (1-a, b) \}$) (Menezes *et al.*, 1996; Lee, 2006) (Table 2). So, the fuzzy operation will be.

Table 2: Fuzzy operation

A	b	Fuzzy
0	0	1
0	1	0
1	0	0
1	1	1

Signal-to-noise ratio: The Signal to Noise Ratio (SNR) is an appropriate yardstick used in the characterization of the physical layer performance. A high SNR at the receiver allows an accurate synchronization. Various modulation formats can be used to exploit the high SNR available at receiver for decoding, thus, enabling high data rates. The reduction in SNR is caused by a long distance between the send and the recipient is define as:

$$SNR = \frac{P_{signal}}{P_{noise}}$$

where, P is represented the average power, since, signals are often Based upon the definition of decibel, so that, signal and noise can be described in decibels (dB) (Menezes *et al.*, 1997; Nelson and Gailly, 1996; Salomon, 2004) as:

$$SNR_{dB} = 10 \log_{10} \left(\frac{P_{signal}}{P_{noise}} \right)$$

Peak signal-to-noise ratio: The Peak Signal-to-Noise Ratio (PSNR) is uses as a measure for ratio between the max (maximum) value for possible power of a signal with the power of sabotage noise which is the light fidelity of its representation, is most easily the PSNR defined via. the Mean Squared Error (MSE) is define the PSNR (in dB) as:

$$PSNR = 10 \cdot \log_{10} \left(\frac{Max_I^2}{MSE} \right)$$

$$= 20 \cdot \log_{10} \left(\frac{Max_I}{MSE} \right)$$

$$= 20 \cdot \text{Log}_{10}(MAX_I) - 10 \cdot \log_{10}(MSE)$$

Here, (MAX_I) is the signal of the maximum possible pixel value (Nelson and Gailly, 1996; Salomon, 2004).

Geffe generator: In this generator uses three LFSR_s the output which is key stream, combined in a non-linear manner (Fig. 2). One is selected and other two are input into multiplexer. If for example, k₁-k₃ are the output of the three LFSR_s, the output of this generator describe as: K(t) = (k₁∧k₃)⊕(k₂∧-k₃).

The period of the Geffe generator is the product of the periodic of the three LFSR_s, by assuming the degree are relatively prime for the three primitive feedback polynomials (Menezes *et al.*, 1997; Van Tilborg, 1998).

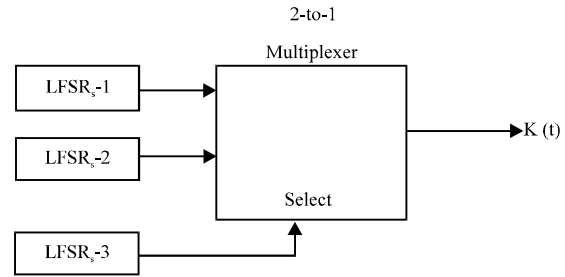


Fig. 2: Geffe generator

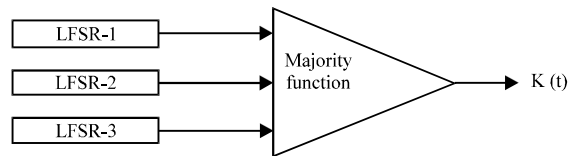


Fig. 3: Threshold generator

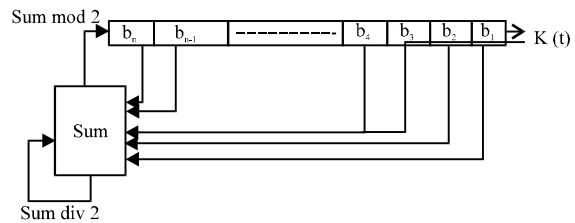


Fig. 4: Feedback with carry shift register

Threshold generator: In this generator which is explaining in the Fig. 3. Take a large number of LFSR_s (by use the odd number) with three LFSR_s as the output, so, the output generator can be written as:

$$K(t) = (k_1 \wedge k_2) \oplus (k_1 \wedge k_3) \oplus (k_2 \wedge k_3)$$

This is seems as the Geffe generator except that it has a large linear complexity of n₁n₂+n₁n₃+n₂n₃. Where the lengths of the LFSR₁-LFSR₃ are n₁-n₃, respectively (Menezes *et al.*, 1997; Van Tilborg, 1998).

Feedback with carry shift register generator: This generator it's seems as the LFSR such that both of them have feedback function and shift register but a Feedback with Carry Shift Register (FCSR) have carry register, so that, instead the XOR_{ing} this generator depend on two steps the first step is summation the tap sequence (bits) with carry register and this result module 2 (Fig. 4) which give us the new bit and same steps with divided by 2 give us new carry (Arnault *et al.*, 2008; Goresky and Klapper, 2002).

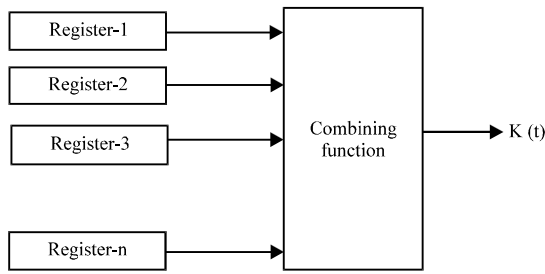


Fig. 5: Combining generators

FCSR combining generators: For this generator use FCSR_s or LFSR_s as a variable numbers and combine them by a variety of function instead of the XOR's operation (Fig. 5) which make the problem with the algebraic properties of FCSR_s, so that, the best it to combine them and the output is the XOR's of the individual FCSR_s (Arnault *et al.*, 2008; Goresky and Klapper, 2002).

RESULTS AND DISCUSSION

Huffman code via. key generator's

Huffman code via. Geffe generator: If the plain text is (IRAQ RIVER) with a four stage LFSR₁-LFSR₃ with the same primitive polynomial $x^4+x+1 = 0$ and initial state (1101), (0011) and (0101), respectively then (Table 3).

Huffman code via. threshold generator: With same plaintext and if take a 4-stage LFSR₁-LFSR₃ with the same tap sequence T = (1001) with initial state (1101), (1110) and (1011), respectively then (Table 4).

Huffman code via. FCSR: With same plaintext and if the initial state 001 with carry register is 0, then (Table 5).

Huffman code via. FCSR combining generator: With same plaintext and if take a 3-stage FCSR₁, FCSR₂ and FCSR₃ with the same carry register is 0 with initial state (001), (011) and (010), respectively then (Table 6-8) and (Fig. 6 and 7).

$K_1 = 100101111000101110001011100$
 $K_2 = 110001011101100010111011000$
 $K_3 = 010111010000101110100001011$
 So that

Measure SNR and PSNR: In this partition, we measure the SNR and PSNR.

Table 3: The same primitive polynomial

Plaintext	Iraq river
Huffman code	010010010101110001101111000
Key stream geffe generator	111000000111110111000000111
XOR's	101010010010000110101111111
Fuzzy	010101101101111001010000000

Table 4: LFSR₁-LFSR₃ with the same tap sequence

Plaintext	Iraq river
Huffman code	010010010101110001101111000
Key stream threshold generator	111100100100110111100100110
XOR's	101110110001000110001011100
Fuzzy	010001001110111001110100011

Table 5: Same plaintext and if the initial state 001 with carry register

Plaintext	Iraq river
Huffman code	010010010101110001101111000
Key stream (FCSR)	100101111000101111000101111
XOR's	110111101101011110101010111
Fuzzy	001000010010100001010101000

Table 6: Stage FCSR1, FCSR2 and FCSR3 with the same carry register

Plaintext	Iraq river
Huffman code	010010010101110001101111000
Key stream (FCSR combining generator)	000011110101100010010001111
XOR's	010001100000010011111101111
Fuzzy	10111001111101100000001000

Table 7: Measure SNR

Operation/Generator types	Fuzzy	XOR
Geffe generator	2.8377	0.6305
Threshold generator	8.0853	-0.4985
FCSR generator	6.7890	-0.2840
FCSR combining generator	1.6594	1.5323

Table 8: Measure PSNR

Operation/Generator types	Fuzzy	XOR
Geffe generator	07.5795	5.3723
Threshold generator	12.8271	4.2433
FCSR generator	11.5308	4.4578
FCSR combining generator	06.4012	6.2741

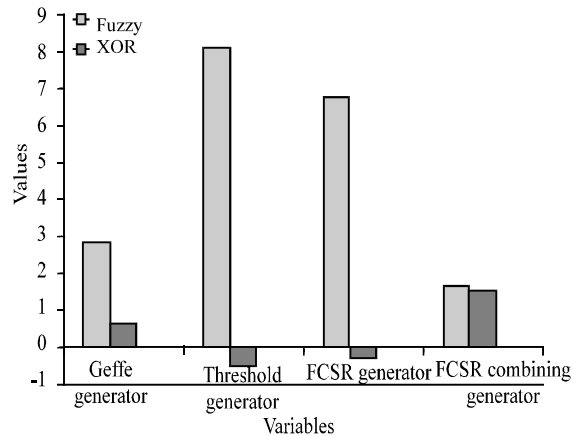


Fig. 6: Generator's type

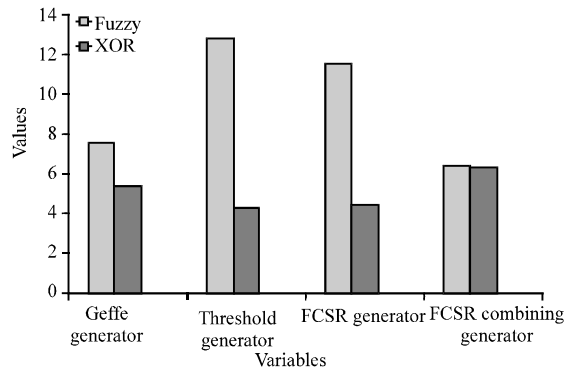


Fig. 7: Generator's type

CONCLUSION

For evaluation measure the cipher's and the difference between measures the SNR and PSNR at least 4.7 for XOR's or fuzzy operations and we get a good result which clear as in the previous diagram.

REFERENCES

Al-Swidi, A. and Y. Al-Talkany, 2014. On adaptive of Generations by using E-A and D-A laws. *Control Theor. Inf.*, 4: 39-52.

Arnault, F., T.P. Berger and M. Minier, 2008. Some results on FCSR automata with applications to the security of FCSR-based pseudorandom generators. *IEEE. Trans. Inf. Theor.*, 54: 836-840.

El Abbadi, N. and E.H. Al-Saadi, 2016. Image de-noising: Comparative study. *J. Babylon University Pure Appl. Sci.*, 24: 1155-1161.

Goresky, M. and A.M. Klapper, 2002. Fibonacci and Galois representations of feedback-with-carry shift registers. *IEEE. Trans. Inf. Theory*, 48: 2826-2836.

Huffman, A., 1952. A method for the construction of minimum-redundancy codes. *Proc. IRE*, 40: 1098-1101.

Lee, C.C.C., 1990. Fuzzy logical in control systems: Fuzzy logic controller-parts I and II. *IEEE Trans. Syst. Man Cybernet.*, 20: 404-430.

Lee, K.H., 2006. *First Course on Fuzzy Theory and Applications*. Springer Science & Business Media, Berlin, Germany, Pages: 335.

Menezes, A.J., P.C. van Oorschot and S.A. Vanstone, 1997. *Handbook of Applied Cryptography*. CRC Press, New York, USA.

Menezes, A.J., P.C.V. Oorschot and S.A. Vanstone, 1996. *Handbook of Applied Cryptography*. CRC Press, Boca Raton, Florida, USA., ISBN-13:978-0-84-938523-0, Pages: 780.

Nelson, M. and J.L. Gailly, 1996. *The Data Compression Book*. M & t Books, New York, USA., ISBN:9781558514348, Pages: 543.

Salomon, D., 2004. *Data Compression: The Complete Reference*. 3rd Edn., Springer, Germany, ISBN-13: 978-0387406978, Pages: 900.

Van Tilborg, H.C., 1998. *Coding Theory at Work in Cryptology and Vice Versa*. In: *Handbook of Coding Theory*, Pless V.S. and W.C. Huffman (Eds.). Elsevier Publishing, New York, USA., ISBN:9780444500878, pp: 1195-1209.