ANALYSIS OF ENCODING PLAINTEXT DATA: USING ENHANCED HAMMING CODE TECHNIQUES

Afolabi Godfrey
PhD Student,
Mathematics Department, Usmanu Danfodiyo University, Sokoto, Nigeria

Garba. A. Isa (PhD)
Head of Department,
Mathematics Department, Usmanu Danfodiyo University, Sokoto, Nigeria

Muhammad .S. Magami, (PhD),
Senior Lecturer,
Mathematics Department, Usmanu Danfodiyo University, Sokoto, Nigeria

Dogondaji. A. M, (PhD),
Senior Lecturer,
Mathematics Department, Usmanu Danfodiyo University, Sokoto, Nigeria
.

ABSTRACT

Most often than none, data (information) obtained as results, which are to be used, saved or transmitted across channels from a source to a specific destination are in form of plaintext. The efficiency and reliability of any result obtained is of great importance to any data (information) analyst (user) and this can only be guaranteed by encoding such results. It is against this background therefore, that this study on Analysis of Encoding Plaintext Data: Using Enhanced Hamming Code Techniques was carried out. More so, the results obtained as shown in this paper shall be useful specifically in plaintext and generally in coding theory.

Keywords: *ASCII, Encode, Parity, Parity key, and Parity key code*.

---

1. INTRODUCTION

Data (information) in form of plaintext; are set of words of letter (character) containing alphabets Aa-Zz, which can either be obtained as results from analysis carried out or data (information) used by data (information) analyst (user) as the case may be[1]. The need to encode such results or data (information) becomes very necessary in other to enhance its efficiency and reliability whenever in use [2]. It is based on this, that this study on Analysis of Encoding Plaintext Data: Using Enhanced Hamming Code Techniques was carried out. The results obtained therefore will be very useful in coding theory.

2. DEFINITION OF ABBREVIATIONS AND BASIC TERMS USED

The following abbreviations and basic terms as used in this paper are defined accordingly to make the work self contained.

i. ASCII: American Standard Code for Information Interchange
ii. DC (Data Code): Taking the hexadecimal equivalence of the bit entries
iii. DP: Data Position
iv. DCM: Decimal
v. Encode: This as the process which involves the addition of parity (correction) bit to the information being sent, stored or computed. This is to enable the identification of error(s) when they occur. Thus, encoding a bit sequence adds redundant information to aid the intended receiver in correcting symbol error(s). For example, to encode the given data 10010011**,** a parity code **1100,** calculated would be imputed in positions 1, 2, 4 and 8 respectively. Thus, the encoded data would be **11**1**0**001**0**0011 [3].
vi. EPTL: Encoded Plaintext Letter
vii. Parity: A binary digit called parity is used to indicate whether the number of bits with '1' in a given set of bits is even or odd, usually used to detect transmission or computation error. The parity bit is then imputed in the original data and does allow for the restoration of an erroneous bit when its position is detected [4].
viii. Parity Key: Taking the reverse of the parity entries.
ix. PKC (Parity Key Code): Taking the hexadecimal equivalence of the parity key entries
x. PTL: Plaintext Letter

3. METHOD / PROCEDURE

The Method of the Analysis of Enhanced Hamming Code Techniques used in this work followed these procedures:

Step i: Get the ASCII character (decimal) equivalence of each letter in the plaintext.

Step ii: Convert the equivalent ASCII character of each letter in the plaintext to binary (8-digits) respectively and place them in data positions 3, 5, 6, 7, 9, 10, 11 and 12 respectively..

Step iii: Generate parity for the binary equivalences of each ASCII character (decimal) of each letter in the plaintext and set their respective parity keys (entries in red), as shown in the TABLE 4.0.1 below [5].

Step iv: Encode each letter in the plaintext by placing the set parity keys (that is the reverse parity) in their respective positions (that is, Data Positions 2^n; n = 0, 1, 2 and 4 respectively: data positions 1, 2, 4 and 8, as the case may be) [6].

Step v: Divide the encoded letter of the plaintext into two (2) blocks, such that, the parity key (that is, **DP** 1, 2, 4 and 8 respectively) forms the first (1st) block, $\mathbf{B^1}$ and the other bits (that is **DP** 3, 5, 6,7, 9, 10, 11 and 12) will form the second (2nd) block, $\mathbf{B^2}$.

Step vi: Take the hexadecimal equivalence of the entries in each block (that is, $\mathbf{B^1}$ and $\mathbf{B^2}$ respectively) to form the parity key code (entries in green) and data code (entries in blue) respectively as shown in the TABLE 4.0.1 below.

Step vii: Concatenate the parity key codes and the data codes to form the encoded plaintext to be used, sent or saved for future use, as the case may be.

4. PRESENTATION OF RESULT AND COMPUTATIONAL ANALYSIS ON THE PLAINTEXT, "Knowledge is Power":

The result of the presentation and computational analysis on the plaintext "Knowledge is Power" as an example is shown in TABLE 4.0.1 below:

Table 4.0.1 Results Obtained from the Computational Analysis on the plaintext "Knowledge is Power"

PTL	ASCII DCM	DP 1	DP 2	DP 3	DP 4	DP 5	DP 6	DP 7	DP 8	DP 9	DP 10	DP 11	DP 12	PKC	DC	EPTL
K	75	1	1	0	0	1	0	0	1	1	0	1	1	0D	4B	0D4B
n	110	1	1	0	0	1	1	0	1	1	1	1	0	0D	6E	0D6E
o	111	1	1	0	1	1	1	0	0	1	1	1	1	0E	6F	0E6F
w	119	1	0	0	0	1	1	1	1	0	1	1	1	09	77	0977
l	108	0	0	0	0	1	1	0	0	1	1	0	0	00	6C	006C
e	101	1	0	0	1	1	1	0	0	0	1	0	1	0A	65	0A65
d	100	1	0	0	0	1	1	0	1	0	1	0	0	09	64	0964
g	103	0	1	0	1	1	1	0	1	0	1	1	1	07	67	0767
e	101	1	0	0	1	1	1	0	0	0	1	0	1	0A	65	0A65
	32	0	1	0	1	0	1	0	0	0	0	0	0	06	20	0620
i	105	0	1	0	1	1	1	0	0	1	0	0	1	06	69	0669
s	115	1	1	0	0	1	1	1	0	0	0	1	1	0C	73	0C73
	32	0	1	0	1	0	1	0	0	0	0	0	0	06	20	0620
P	80	0	1	0	0	1	0	1	0	0	0	0	0	04	50	0450
o	111	1	1	0	1	1	1	0	0	1	1	1	1	0E	6F	0E6F
w	119	1	0	0	0	1	1	1	1	0	1	1	1	09	77	0977
e	101	1	0	0	1	1	1	0	0	0	1	0	1	0A	65	0A65
r	114	1	1	0	1	1	1	1	1	0	0	1	0	0F	72	0F72

Source: Researcher's Calculations

Therefore, the encoded plaintext "Knowledge is Power", which could be used, sent or saved for future use, is given as: "**0D4B0D6E0E6F0977006C0A65096407670A65062006690C73062004500E6F09770A650F72**"

5. CONCLUSION

In general, the efficiency and reliability of data (information) obtained as results, which may either be for the present or future use, is of great importance to any data (information) analyst (user). This can only be guaranteed by encoding such results. It is against this background therefore that this work was carried out. Although the Analysis of Encoding Plaintext Data: Using Enhanced Hamming Code Techniques as presented in this work was carried out on the plaintext "Knowledge is Power", the results obtained however can be applied specifically on all plaintext data (information) and coding theory in general.

REFERENCES

[1] Shweta, S. & Amita, S. *Analysis of EnDeCloudReports for Encrypting and Decrypting Data in Cloud. International Journal of Computer Applications (0975 – 8887). Volume 136 – No. 12, February 2016*.

[2] Isnar, S, Andysah, P. U. S & Arpan, *Base64 Character Encoding and Decoding Modeling, International Journal of Recent Trends in Engineering & Research (IJRTER) Volume 02, Issue 12; December – 2016 [ISSN: 2455 – 1457]*.

[3] Ziegler, J. F, An M.Sc Thesis submitted to the faculty of Information Technology and Engineering of George Mason University. *Automatic Recognition and Classification of Forward Error Correction Code. (Spring 2000, George Masson University Fairfax, Virginia)*.

[4] Bhattacharryya, D. K and Nandi S, (1997) *An efficient class of SEC-DED-AUED codes: International symposium on parallel Architectures, Algorithms and Networks (ISPAN). 1, 410-415*.

[5] Afolabi, G. & Ibrahim, A. A. *The use of Algorithmic Method of Hamming Code Techniques for the Detection and Correction of Computational Errors in a Binary Coded Data: Analysis on an Integer SequenceA119626; IOSR Journals of Mathematics (IOSR- JM) e – ISSN: 2278, P – ISSN: 2319 – 7676. Volume 9, Issue 2 (Nov. – Dec. 2013) pp 33 – 37. www.iosrjournals.org*.

[6] Afolabi, G, Ibrahim, A. A & Zaid, I, *The use of Computational Method of Hamming Code Techniques for the Detection and Correction of Computational Errors in a Binary Coded Data: Analysis on an Integer SequenceA119626; International Journal of Computational Engineering Research (IJCER). ISSN: 2250 – 3005. Volume 04, Issue 1 (January 2014) 6 – 15. www.ijcer.org.*