A New Approach for Multiple Error Detection and Multiple Error Correction of Plaintext Data: Using Enhanced Hamming Code Techniques

Afolabi Godfrey
(Ph.D. Student, Mathematics),
Mathematics Department, Usmanu Danfodiyo University, Sokoto, Nigeria

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

ABSTRACT: *In recent times, data (information) analysts (users) are faced with several security-related issues during data (information) processing; computation, transmission, and storage as the case may be. In a bid to enhance the reliability and efficiency of data (information) whenever in use, encryption and decryption of such data (information) becomes necessary. More so, since the occurrence of errors during data (information) processing cannot be ruled out, this paper, therefore, seeks to present a new approach for multiple error detection and multiple error correction of plaintext by using enhanced hamming code techniques provided the errors occurred in different blocks. The results obtained as presented in this paper would be very useful specifically in the area it was computed and generally in the knowledge of coding theory.*

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

1. INTRODUCTION

According to [1], plaintext data (information) are a set of words of letters (character) containing alphabets in (upper or/and lower cases) Aa – Zz. Plaintext data (information) could either be obtained as results from the analysis carried out or inputs used by data (information) analysts (users) as the case may be [2]. In other to enhance the security of data (information), which is paramount to every data (information) analyst (users), encryption and decryption of such data (information) are carried out [3]. However, to ascertain the reliability and efficiency of data (information) whenever in use, detection and correction of errors which is/are inevitable during data (information) processing becomes very necessary [4]. It is against this background, therefore, that this new approach for multiple error detection and multiple error correction of plaintext data: using enhanced hamming code techniques was carried out. The results of this work as presented in this paper will therefore be of great improvements on the existing hamming code techniques and a plus to the general knowledge of coding theory.

2. DEFINITION OF ABBREVIATIONS AND BASIC TERMS

In other to make this work self-contained, the abbreviations and basic terms used are defined accordingly:

i. ASCII: American Standard Code for Information Interchange

ii. DC (Data Code): Taking the hexadecimal equivalence of the bit entries

iii. DCM: Decimal

iv. Decrypt: This involves the procedures and process of getting back the original data (information) from the encrypted data (information).

v. DP: Data Position

vi. DPTL: Decrypted Plaintext Letter

vii. EBP: Erroneous Bit Position

viii. Encrypt: This is the process that 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, encrypting 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 encrypted data would be **11**1**0**001**0**0011 [5].

ix. EPTL: Encrypted Plaintext Letter

x. 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 [6].

xi. Parity Key: Taking the reverse of the parity entries.

xii. PKC (Parity Key Code): Taking the hexadecimal equivalence of the parity key entries

xiii. PTL: Plaintext Letter

3. METHOD / PROCEDURE

This method of the new approach of multiple error detection and multiple error correction of plaintext using enhanced hamming code techniques followed the following procedures:

Stage I: Verification Analysis

Step 1: Compare the encrypted (original or sent) data (information) with the decrypted (received or retrieved) data (information).

Step 2: If they are the same, then accept, use or save the data (information) and end the analysis. But if otherwise, then proceed to the next stage.

Stage II: Detection Analysis

Step 1: Divide the decrypted (received) plaintext data (information) two (2) blocks **B1** and **B2** respectively; such that, the parity key code (PKC) is in **B1** while the data code (DC) is in **B2**.

Step 2: Take and place the binary (4-digits for PKC and 8-digits for DC) equivalences of entries in **B1**: **DP** 2^n; n = 0, 1, 2and 3 (that is positions 1, 2, 4 and 8 respectively) and **B2**: **DP** 3, 5, 6, 7, 9, 10, 11 and 12 respectively.

Step 3: Take the decimal equivalence of the data code (DC), that is, entries in **DP** 3, 5, 6, 7, 9, 10, 11, and 12 respectively.

Step 4: Take the ASCII character (alphabet/letter) of the respective decimal equivalences of the data code (DC).

Step 5: Concatenate the ASCII character (alphabet/letter) in each block to form the decrypted (decoded) plaintext data (information), retrieved at the receiver's end.

Stage III: Correction Analysis

Step 1: Place each character of the decrypted plaintext data (information) into blocks then take their respective ASCII decimal and binary equivalences.

Step 2: Set the parity key and form the parity key codes of each block.

Step 3: Compare and set the parity key codes of the encrypted (original) data (information) with that of the decrypted (received or retrieved) data (information) to ascertain the deviated block and the erroneous bit(s) in it [7].

Step 4: Correct the erroneous bit(s) in the deviated block by flipping it, that is, change it from '**0**' to '**1**' or from '**1**' to '**0**' as the case may be [5].

4. PRESENTATION OF THE ANALYSIS OF MULTIPLE ERROR DETECTION AND MULTIPLE ERROR CORRECTION (MEDMEC) OF THE ENCRYPTED AND DECRYPTED PLAINTEXT, "Knowledge is Power"

Suppose the plaintext "Knowledge is Power" as an example is encrypted as "**0D4B0D6E0E6F0977006C0A6509640767 0A65062006690C73062004500E6F09770A650F72**", [1] and decrypted as "**Knowledge is Power**", [2]. Then the plaintext data (information) would be accepted for use, transmission, or stored for future use as the case may be, since they are the same.

But if otherwise, say, at the receiver's end, the encrypted plaintext was "**04430D6E0E6F09770568 0A6509640767 0475062006690F72062004500E6F09770767 0F72**" as against "**0D4B0D6E0E6F0977006C0A6509640767 0A65062006690C73062004500E6F09770A650F72**", and decrypted as "**Cnowhedgu ir Powgr**" as against "**Knowledge is Power**". Then, the deviated blocks where the erroneous bit(s) is and their respective positions are identified, and thereafter the erroneous bit(s) corrected by flipping; either from "**0" to "1**" or from "**1" to "0**" as the case may be and as shown in TABLE 4.0.1 below:

Table 4.0.1 Results of Multiple Error Detection Analysis Obtained from Computation on encrypted erroneous plaintext data (information)
"**0D4B0D6E0E6F0977006C0A6509640767 0A65062006690C73062004500E6F09770A650F72**" as "**Knowledge is Power**" being
"**04430D6E0E6F09770568 0A6509640767 0475062006690F72062004500E6F09770767 0F72**" as
"**Cnowhedgu ir Powgr**"

EPTL	PKC	DC	DP 1	DP 2	DP 3	DP 4	DP 5	DP 6	DP 7	DP 8	DP 9	DP 10	DP 11	DP 12	ASCII DCM	DPTL	EBP	Flip EBP From	Flip EBP To
0443	**04**	**43**	0	1	0	0	1	0	0	0	0	0	1	1	**67**	**C**	**9**	**0**	**1**
0D6E	0D	6E	1	1	0	0	1	1	0	1	1	1	1	0	11O	n	-	-	-
0E6F	0E	6F	1	1	0	1	1	1	0	0	1	1	1	1	111	o	-	-	-
0977	09	77	1	0	0	0	1	1	1	1	0	1	1	1	119	w	-	-	-
0568	**05**	**68**	0	1	0	0	1	1	0	1	1	0	0	0	**104**	h	**10**	**0**	**1**
0A65	0A	65	1	0	0	1	1	1	0	0	0	1	0	1	101	e	-	-	-
0964	09	64	1	0	0	0	1	1	0	1	0	1	0	0	100	d	-	-	-
0767	07	67	0	1	0	1	1	1	0	1	0	1	1	1	103	g	-	-	-
0475	**04**	**75**	0	1	0	0	1	1	1	0	0	1	0	1	**117**	**u**	**7**	**1**	**0**
0620	06	20	0	1	0	1	0	1	0	0	0	0	0	0	32		-	-	-
0669	06	69	0	1	0	1	1	1	0	0	1	0	0	1	105	i	-	-	-
0F72	**0F**	**72**	1	1	0	1	1	1	1	1	0	0	1	0	**114**	r	**12**	**0**	**1**
0620	06	20	0	1	0	1	0	1	0	0	0	0	0	0	32		-	-	-

0450	04	50	0	1	0	0	1	0	1	0	0	0	0	0	80	P	-	-	-
0E6F	0E	6F	1	1	0	1	1	1	0	0	1	1	1	1	111	o	-	-	-
0977	09	77	1	0	0	0	1	1	1	1	0	1	1	1	119	w	-	-	-
0767	**07**	**67**	0	1	0	1	1	1	0	1	0	1	1	1	**103**	**g**	**11**	**1**	**0**
0F72	0F	72	1	1	0	1	1	1	1	1	0	0	1	0	114	r	-	-	-

Source: Researcher's Calculations

Therefore, the corrected erroneous decrypted plaintext data (information) "**04430D6E0E6F09770568 0A6509640767 0475062006690F72062004500E6F097707670F72**" being "**Cnowhedgu ir Powgr**" which could be used, sent or saved for future use, as the case may be, is given as: "**0D4B0D6E0E6F0977006C0A6509640767 0A65062006690C73062004500E6F09770A650F72**" being "**Knowledge is Power**" as shown in TABLE 4.0.2 below:

Table 4.0.2 Results of Multiple Error Correction Analysis Obtained from Computation on decrypted erroneous plaintext data (information) "04430D6E0E6F09770568 0A6509640767 0475062006690F72062004500E6F097707670F72" as "Cnowhedgu ir Powgr" being "0D4B0D6E0E6F0977006C0A6509640767 0A65062006690C73062004500E6F09770A650F72" as "Knowledge is Power"

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

Source: Researcher's Calculations

5. CONCLUSION

Data (information) security can only be fully realized when a mechanism for detection and correction of computational or/and transmission error(s) which could occur during data (information) processing, is put in place. Thus, the results of this work as presented in this paper, showing a new approach of multiple error detection and multiple error correction of plaintext data by using enhanced hamming code techniques will be great improvements on the existing hamming code techniques and a tremendous contribution to general knowledge of coding theory.

REFERENCES

[1] Afolabi. G, Garba. A. I, Muhammad. S. M and Dogondaji. A. M, "*Analysis of encoding plaintext data: using enhanced hamming code techniques"; International Journal of Advanced Research and Innovative Ideas in Education (IJARIIE). ISSN (O)-2395-4396. Vol-7, Issue-5, 2021. www.ijariie.com*

[2] Afolabi. G, Abba. A and Dogondaji. A. M *'Analysis of Decoding Plaintext Data: Using Enhanced Hamming Code Techniques" International Journal Of Trend in Scientific Research and Development (IJTSRD)Volume 6 Issue 1, November-December 2021.www,ijtsrd.com, e-ISSN: 2456 – 6470*

[3] 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*.

[4] 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].*

[5] 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.*

[6] 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.*

[7] 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*.