Article can be accessed online at http://www.publishingindia.com

An Enhanced UPC Mod 10 Check Digit Method

Zahraddeen A. Pindar* , Sapiee Jamel, Abdulkadir H. Disina*****

Abstract

The advent of check digit methods has aided the detection of errors while computing data into an information system. Prominent check digit methods such as Universal Product Code (UPC), International Standard Book Number (ISBN-10), Luhn and Verhoeff check digit methods are widely used in detecting errors. This research will focus on the current UPC method as it is used widely around the world. The current UPC method does not effectively detect all types of errors. Thus, failure to detect errors can lead to undesired consequences like purchasing a wrong item. This paper examines the limitations of the UPC check digit method and proposes the idea of a double check digit method which produces two check digits. "....The proposed method is an enhancement of the current UPC modulus 10 check digit method, which is based on two check digits. The two check digits are generated separately. A comparative text was conducted". A comparative test was conducted to compare the ability of the proposed method and the current UPC method in detecting errors. However, results have shown that the proposed method has an average of 95.75 accuracy of detecting all tested errors as compared to the current UPC which detects 65% of tested errors respectively. Thus, the proposed method is more efficient than the current UPC check digit method in detecting errors.

Keywords: Double Check Digit, Double Check Digits, Errors

1. Introduction

Integrity is one of the most fundamental security properties (Khan, 2013). Integrity protection mechanisms can be grouped into two mechanisms: prevention and detection mechanism. Prevention mechanism prevents unauthorised individuals from modifying the document. Detection mechanism detects unauthorised modification when preventive mechanism has failed. Methods such as check digits, checksum, digital signatures and hash functions are detective mechanisms that can be used to detect unauthorised changes (Chen et al., 2012; Damm, 2000). Check digit has been in use extensively around the world for a long time. Other well-known examples are check digits used in International Bank Account Number (IBAN) in bank account number used in many countries (Bahrain, 2011 & Pakistan, 2013), International Standard Book Number (ISBN) numbers of books and UPC found in groceries (Khaled, 1998; Kamaku et al., 2012; Siddharth, 2012). Check digit methods can be categorized into single error and double error check digit. The two categories are standardized under ISO. Research in (Khaled, 1998) indicates that errors which occur in machine communication are caused by human operations. For instance, in the process of keying-in codes into a database manually to store or retrieve information, while doing so errors do often occur and this may lead to many unwanted outcomes. A practical example is when a wrong UPC code is entered; this may result in purchase of an undesired book. Hence, it is important that these errors are detected and users are notified. It is for this significant reason that led to the design of error detection methods; check digit methods. These check digit methods are designed to detect errors which occur while data is being keyed into an information system.

* Department of Information Security, Universiti Tun Hussein Onn Malaysia, Batu Pahat, Malaysia. Email:deenpindar@gmail.com

Overview of Common Errors

Based on an empirical investigation by (Verhoeff, 1969) and according to Damm (2000) and Gallian (1996), six (6) types of prominent errors occur while data is keyed into a system. data is keyed into a system. The errors are as follows:

Single Digit Error

This error occurs when only a single digit in the whole string of numbers or alphabets is typed wrongly.

Example 1

123456 is typed as 123356. Here the digit 4 is typed as 3.

Single Transposition Error

This type of error occurs when the nth and the (n+1) thdigits are replaced by the (n+1)th and the nth respectively.

Example 2

123456 typed as 126453. Here the digits 3 and 6 are transposed.

Jump Transposition Error

This type of error occurs when the nth and the (n+2)th digits are replaced by the (n+2)th and the nth respectively.

Example 3

123456 typed as 123654. Here the digits 4 and 6 are jump transposed.

Twin Error

This type of error occurs when two consecutive identical digits are incorrectly typed as two other identical digits.

Example 4

124456 typed as 129956. Here the digits 44 are incorrectly typed as 99.

Jump Twin Error

This type of error occurs when the nth and (n+2)th digits are identical and are mistyped as two other identical digits.

Example 5

123454 mistyped as 123858. As 454 is jumped transposed to be 858.

Phonetic Error

This type of error occurs when 15 (fifteen) is heard as 50 (fifty). It can also be noted that more than one type or error can occur at a time.

Table 1 gives a summary of a large experiment carried out on types of errors and their pattern and the frequency of their occurrences.

Table 1: Prominent Errors and Their Frequency of Occurrences

Error type	*Form*	*Relative frequency*
Single error	$a \to b$	79.1%
Single transposition error	$ab \to ba$	10.2%
Jump transposition	$abc \to cba$	0.8%
Twin error	$aa \to bb$	0.5%
Phonetic error	$a0 \to 1a$	0.5%
	a= 2,…. , 9	
Jump twin error	$aca \to bcb$	0.3%

(Gallian, 1996; Kamaku et al., 2012" Siddharth, 2012)

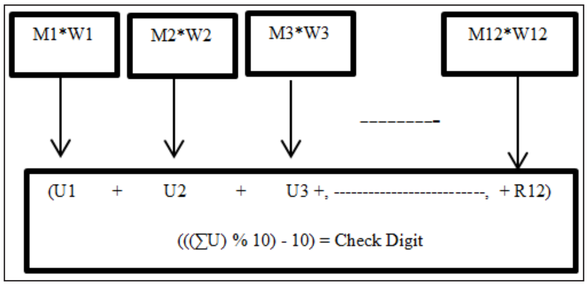
UPC Modulus 10 Method

The UPC (Universal Product Code) is a standard coding scheme used to uniquely identify products. It consist of a twelve digits numeric code and one digits appended to it, known as the check digit (Gallian, 1991; Gumm, 1985). The first digit identifies a broad category of product type, the next five digits identify the manufacturer, the next five the product, and the last is the check digit. The UPC code is represented by M_1, M_2… M_{12}, the check digit M_{13} is appended to the code.

$$U = \sum_{i=1}^{n=12} (Mi * Wi) \qquad 1.1$$

where weight W = (3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1) and UPC number M = (M1, M2… M12). Fig.1 represents a diagrammatic view of the UPC method.

Fig. 1: Diagrammatical Representation of UPC Modulo 10 Check Digit Method



where weight is W1-12 and UPC number is M1-12.

Example 6

Say181100320763 is the UPC number of a product. Let's compute the check digit.

Fig. 1(a): Computation of Check digit

1*	8*	1*	1*	0*	0*	3*	2*	0*	7*	6*	3*	=56
3	1	3	1	3	1	3	1	3	1	3	1	

54 % 10 = 4, 10 – 4 = 6, therefore 6 is the check digit. However, the problem of the current UPC method is that it allows all jump transposition errors to go undetected. In Fig. 1(b) 2 & 0 and 3 & 7 are transposed.

Fig. 1(b): Jump Transposition Error

1*	8*	1*	1*	0*	2*	3*	0*	0*	3*	6*	7*	=54
3	1	3	1	3	1	3	1	3	1	3	1	

54 % 10 = 4, 10-4 = 6.

When the digits are computed, the resulting check digit does not change. Hence, this indicates the method does not detect jump errors or multiple jump errors. The ineffectiveness of the method to detect all errors motivated this research to device a means that will detect all errors. The next section of this paper will discuss the proposed method.

Proposed Method

This section will discuss the methodology of the double check digit. The proposed double check digit method is presented in the diagram below for better illustration. The procedure for computing the 1st check digit will be discussed first, and in later section of this paper, the procedure for computing the second check digit will also be discussed.

Computing 1st Check Digit

The first code digit M1 is multiplied with weight W1, the result is denoted as (M1 * W1). The result is then XOR with (M12 * W12) to produce the first intermediate result "X1". The second digit code M2 is also multiplied with its weight W2, the result is denoted as (M2 * W2). To produce the second intermediate result "X2", the result is then XOR with (M1 * W1) i.e. (M1 * W1) ⊕(M2 * W2). This process is continued until the 12^{th} code digit "X12" i.e (M11 * W11) ⊕(M12 * W12). The intermediate results (X1, X2, X3… X12) are summed up and modulu with 10 (ten), and then 10 (ten) is subtracted from the result. This would produce the first check digit. It is denoted as (∑X % 10) – 10 = CD 1. Hence interdependence is created between code digits. Fig. 2 represents a diagrammatical representation of the method described above.

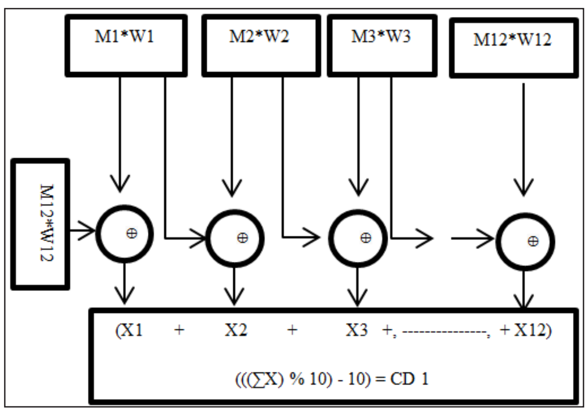
The equation below represents a mathematical method for computing the 1^{st} check digit.

$$X = (M12 * W12) \oplus (M1 * W1)$$

$$\sum_{i=1}^{n=11} (Mi * Wi)$$

$$\oplus (M[i+1] * W[i+1]) \qquad 1.2$$

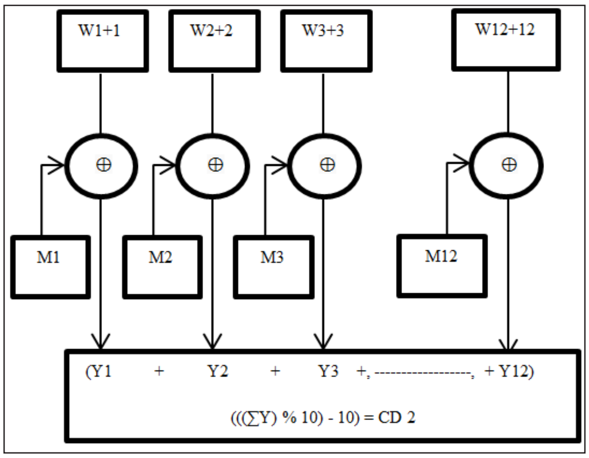
Fig. 2: Diagrammatical Representation of 1st Check Digit Method



Computing 2nd Check Digit

The method of computing the 2nd check digit is different from the 1st method. In computing the 2nd check digit, the same code word and weight used in the 1st method will be used here. 1 (one) is added to the first weight W1, and the sum is XOR with M1. This produces the first intermediate resultfor this method "Y1". The result is denoted as M1⊕ (W1+1). 2 (two) is added to the second weight W2, and the sum is XOR with M3 to produce the second intermediate result "Y2". The result is denoted as M2⊕ (W2 + 2). The same procedure is continued until the 12th code digit "Y12" i.e M12⊕ (W12 + 12). The intermediate results here (Y1, Y2, Y3… Y12) are also summed up. The summed up result is modulu with 10 (ten), and then 10 (ten) is subtracted from the result. This would produce the first check digit. It is denoted as (∑Y % 10) - 10 = CD 1. However, interdependence between codes is not created is it was done in the first method. Fig. 3 below represents a diagrammatical representation of the method described above.

Fig. 3: Diagrammatical Representation of 2nd Check Digit Method



The equation below represents a mathematical method for computing the 2nd check digit.

$$Y = \sum_{i=1}^{n=12} ((Mi \oplus (Wi + i)) \qquad 1.3$$

Algorithm Pseudocode

This subsection describes the algorithm for the proposed methodology.

BEGIN
INITIALIZE all arrays and variables
FOR i = 0 to 11
COMPUTE X[i] as XOR of M[i] multiplied With W[i] and M [i+1] multiplied WithW [i+1]
INCREMENT i
END FOR
FOR i=0 to 11
COMPUTE ResultCD1 as sum of all Columns in array X [i]
INCREMENT i
END FOR
COMPUTE ResultCD1 as sum of ResultCD1 and X [0]
COMPUTE ResultCD1 as ResultCD1 modulus 10
SUBTRACT 10 from ResultCD1
FOR i=0 to 11
COMPUTE Y[i] as sum of W[i] and [i] XOR M[i]
INCREMENT i
END FOR
FOR i=0 to 11
COMPUTE Result CD2 as sum of all columns in array Y[12]
INCREMENT i
END FOR
COMPUTE Result CD2 as Result CD2 modulus 10
SUBTRACT 10 from Result CD2
CONCATINATE Result CD1 and Result CD2 as Double Check Digit

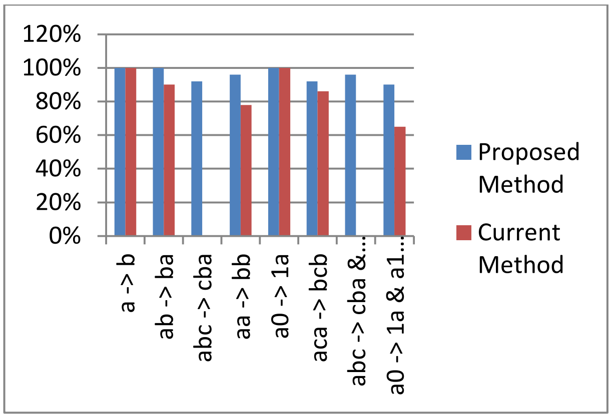
Result of Proposed Methodology

This section discusses the result generated from testing the two check digit methods (current UPC and proposed method). Analysis of result is described in detail.

Graphical Representation of Test Result

Fig. 4 shows a comparative test result of the two methods. 450 test data were used in the comparative testing. The test data was generated randomly. Each test-data contains 12 digits. This is done in-order to meet the UPC standard digit length. The test result is analyzed further in the next subsection.

Fig. 4: Test Result of Two Methods



Analysis of Test Result

Table 2: Analysis of Result Experiment in Tabular Form

Types of Errors	*Proposed Method*	*Current Method (UPC)*
Single Error	100%	100%
Single Transposition Error	100%	90%
Jump Transposition Error	92%	0%
Twin Error	96%	78%
Phonetic Error	100%	100%
Jump Twin Error	92%	86%
Double Jump Transposition Error	96%	0%
Double Phonetic Error	90%	82%
Total Accuracy	95.75%	65%

Gallian (1996) argued that more than one error can occur at a given time. Based on that, we introduced two more types of error namely; double jump transposition error and double phonetic error to test the two method. Double jump transposition error is based on the idea that two jump transposition errors can occur at a time, and the double phonetic error is also based on idea that two phonetic errors can occur at a time.

The current UPC method based on the test result detects 100% of single and phonetic errors. It detects 90%, 78%, 86% and 82% of single transposition error, twin error, jump twin error and double phonetic error respectively and 0% of jump transposition and double jump transposition error. Table 2 gives a detail summary of the test result.

Based on the above analysis, our proposed method proves to be more efficient than the current UPC method.

Conclusion and Future Work

Check digit methods are used to detect errors while keying-in data into an information system. Inability to detect errors can lead to undesired consequences like purchasing a wrong item. The limitation of UPC check digit method has been examined in this paper. A new check digit method is proposed. The proposed method is an enhancement of the current UPC check digit method.

Two new types of errors were introduced in the testing; double jump transposition error and double phonetic error to test the two methods.

A comparative test was carried out to compare the efficiency of the two methods (current UPC and proposed method) in detecting errors. The proposed method has a total accuracy of 95.75% and the UPC check digit method has a total accuracy of 65%. The test results have shown that the proposed method is more efficient than the current UPC method in error detection.

Schulz (1991) outlines that check digit systems based on Latin squares are to some extent better than that of a modulo m check digit system. In line with this argument, the proposed method can be further enhanced or modified by integrating into it more mathematical functions and properties such as permutations and quasi-groups to improve its error detection capability.

References

1. Chen, Y., Niemenmaa, M., Vinck, A. J. & Gligoroski, D. (2012). *On Some Properties of a Check Digit System*. In Proceedings of IEEE International Symposium on Information Theory (pp. 1563-1567).
2. Damm, M. (2000). Check digit systems over groups and anti-symmetric Mappings. *Archivder Mathematick*, 75(6), 413-421.
3. Gallian, J. A. (1996). Error detection methods. *ACM Computing Surveys*, 28(3), 504.
4. Gallian, J. A. (1991). The mathematics of identification numbers. *College Mathematics Journal*, 22(3), 194-202.

5. Guidelines: IBAN Implementation in Pakistan (2012). Retrieved from http://www.sbp.org.pk. (accessed on September 29, 2013).
6. International Bank Account Number (IBAN) Standard for the Kingdom of Bahrain (2011). Retrieved from http://www.cbb.gov.bh. (accessed on September 29, 2013).
7. Khaled, A. S. (1998). Detecting substitutions and transpositions of characters. *The Computer Journal*, 41(4), 270-277.
8. Khan, Q., Butt, M. A., Zaman, M. & Asger, M. (2013). A novel approach based information integrity modeling. *International Journal of Engineering Science & Innovative Technology*, 2(1), 210-215.
9. Kamaku, W., Mwathi, C. & Kivunge, B. (2012). Limitations inthe convectional ISBN-10 code. *American International Journal of Contemporary Research*, 2(2), 150-152.
10. Siddharth, A. (2012). Error Detection in Numeric Codes, 17(7), 653. Resonance-Heidelberg.
11. Verhoeff, J. (1969). Error Detecting Decimal Codes. The Mathematical Centre. Amsterdam.