On the Weaknesses in Error Detection and Correction in the ISBN-13

Waweru Kamaku

Department of Pure and Applied Mathematics, Jomo Kenyatta University of Agriculture and Technology. P.O Box 62000-00200 Nairobi Kenya

Abstract: ***The ISBN-13 uniquely identifies books published internationally since 2007. Kamaku (2012) discussed on key properties of this code as far as error detection is concerned. This paper extends their work and analyses the error detection and correction capabilities of this code and goes on discuss major weaknesses in both error detection and correction and then shows that the code does not guarantee both.***

Keywords: Code, dictionary, ISBN-13, error detection, error correction.

Propositions 1

The lower limit of the number of bit strings that can be in error in an ISBN-13 codeword to yield to a silent error is 2 .

Proof. Since the number of bit strings cannot be negative, then we only need to show that it cannot be zero or one. Suppose no bit string is in error in a valid code word. Thus the code word remains the same and similarly no silent error. If one bit string is in error, the check digit evaluation method would detect the error thus no silent error. Hence the lower limit on the number of bit strings that can be in error to yield to a silent error is 2.

This proof leads to the need of determining the upper limit on the maximum number of errors on a code word that can occur to yield to a silent error.

Theorem 1

In an ISBN-13 code, any multiple silent errors on even positions of a sent code word that yields same weighted sum modulo 10 cannot be detected nor corrected.

Proof: Suppose there are n silent errors on n even positions of the code word $C_1 = a_1 a_2 \cdots a_{12} a_{13}$ to yield to a code word $C_2 = b_1 b_2 \cdots b_{12} b_{13}$ which differ with C_1 in n bit strings such that each even position a_i, in is replaced by b_i for $1 \le 2i \le 12$. Upon calculation of the check digit, the even positions are each multiplied by 3.

Suppose $3\sum_{2i}^{12} a_i \equiv c$ (mod 10), for $0 \le i \le 6$ and $3\sum_{2i}^{12} b_i \equiv c$ (mod 10) for $0 \le i \le 6$ (that is, they yield the same weighted sum modulo 10).

Then $(3\sum_{2i}^{12} a_i)$ - c= $10\ k_1$, $k_1 \in \mathbb{Z}$ and $(3\sum_{2i+1}^{12} b_i)$ - $c = 10\ k_2$, $k_2 \in \mathbb{Z}$.

Subtracting yields $3\sum_{2i}^{12} a_i$ - $3\sum_{2i+1}^{12} b_i = 10\ (k_1 - k_2)$. Thus $\sum_{2i}^{12} a_i \equiv \sum_{2i+1}^{12} b_i$ (mod 10).

Any interchange of a_i with b_i, for $0 \le i \le 6$, satisfying this equation will be an error which would go unnoticed.

Example 2.2.5.1 Suppose the code word ***a*** = 9780198538035 is sent and suppose three errors occur on the second, fourth and sixth bit strings to yield a code word received as ***b*** = 9881178538035. As discussed earlier, 9780198538035 is a valid ISBN-13 code. Considering 9881178538035

s = 9×1 + 8×3 + 8×1 + 1×3 + 1×1 + 7×3 + 8×1 + 5×3 + 3×1 + 8×3 + 0×1 + 3×3
= 9 + 24 + 8 + 3 + 1 + 21 + 8 + 15 + 3 + 24 + 0 + 9 = 125 ≡ 5 mod 10.
But 10 – 5 = 5 ≡ 5 mod 10 thus our check digit is 5. Hence 9881178538035 is also a valid code word. This means that even though errors occurred during transmission, the syndrome will not detect the error made!

Corollary 1
In an ISBN-13 code, any number of silent errors on odd positions on a sent code word that yields same weighted sum modulo 10 cannot be detected nor corrected.

Proof: Suppose there are n silent errors on n odd positions of the code word $C_1 = a_1 a_2 \cdots a_{12} a_{13}$ to yield to a code word $C_2 = b_1 b_2 \cdots b_{12} b_{13}$ which differ with C_1 in n bit strings such that each odd positions, a_i, in error is replaced by b_i for $1 \le 2i+1 \le 11$. Upon calculation of the check digit, the odd positions are each multiplied by 1.

Suppose $\sum_{2i+1}^{11} a_i \equiv c\ (mod\ 10)$ and $\sum_{2i+1}^{11} b_i \equiv c\ (mod\ 10)$ where $0 \le i \le 5$.

Then $(\sum_{2i+1}^{11} a_i)$ - c= $10\ k_1$, $k_1 \in \mathbb{Z}$ and $(\sum_{2i+1}^{11} b_i) - c = 10\ k_2$, $k_2 \in \mathbb{Z}$. Subtracting yields
$\sum_{2i+1}^{11} a_i$ - $\sum_{2i+1}^{11} b_i = 10\ (k_1 - k_2)$. Thus $\sum_{2i+1}^{11} a_i \equiv \sum_{2i+1}^{11} b_i (mod\ 10)$

Any interchange of a_i with b_i, for $0 \le i \le 5$, satisfying this equation will be an error which would go unnoticed.

Example 2.2.5.2.1Consider the sent code word 9780198538035, received code word 9740598538035. As discussed earlier, 9780198538035 is a valid ISBN-13 code. Considering 9740598538035
s = 9×1 + 7×3 + 4×1 + 0×3 + 5×1 + 9×3 + 8×1 + 5×3 + 3×1 + 8×3 + 0×1 + 3×3

= 9 + 21 + 4 + 0 + 5 + 27 + 8 + 15 + 3 + 24 + 0 + 9 = 125 ≡ 5 mod 10.

But 10 – 5 = 5 ≡ 5 mod 10 thus our check digit is 5. Hence 9740598538035 is also a valid code word. This means that even though errors occurred during transmission, the syndrome will not detect the errors!

Corollary 2
The upper limit on the maximum number of errors on a code word that can occur to yield to a silent error is 12.

Proof: The check digit is computed from the other digits and as shown above in Corollary 1 any multiple silent errors on a sent code word that yields the same weighted sum modulo 10 cannot be detected nor corrected, hence the upper limit is 12.

Propositions 2 In an ISBN-13 codeword, If the following values on bit strings are replaced with the ones indicated with an "or", then a silent double error occurs where $1 \leq i \leq 12, 1 \leq j \leq 12,$

i. $a_i = 1, a_j = 9$ or $a_i = 2, a_j = 8$ or $a_i = 3, a_j = 7$ or $a_i = 4, a_j = 6$ or $a_i = 5, a_j = 5$
ii. $a_i = 2, a_j = 9$ or $a_i = 3, a_j = 8$ or $a_i = 4, a_j = 7$ or $a_i = 5, a_j = 6$
iii. $a_i = 3, a_j = 9$ or $a_i = 4, a_j = 8$ or $a_i = 5, a_j = 7$ or $a_i = 6, a_j = 6$
iv. $a_i = 4, a_j = 9$ or $a_i = 5, a_j = 8$ or $a_i = 6, a_j = 7$
v. $a_i = 5, a_j = 9$ or $a_i = 6, a_j = 8$ or $a_i = 7, a_j = 7$
vi. $a_i = 6, a_j = 9$ or $a_i = 7, a_j = 8$
vii. $a_i = 8, a_j = 8$ or $a_i = 9, a_j = 7$

The proof is as a consequence of findings by(Kamaku et al., 2012)since they leave the same weighted sum modulo 10. They proved that

In an ISBN-13 code, any double or multiple silent errors on odd positions of a sent code word that yields same sum modulo 10 with the ones on the received code word cannot be detected nor corrected (Kamaku et al., 2012, p162)

Proposition 3
The ISBN-13 can detect any single error on any even digit position.

Proof: By contradiction, suppose it cannot. This means that there exists two code words **x** and **y** which differ on one even digit position say *i*. Since *i* is even, then *i*= 2n, n ∈ ℤ, 1 ≤ n ≤ 6. Let c_i and d_i be bit strings on **x** and **y** respectively at which they differ. The check digit is not an even position thus it is not in error and hence the check digits for the code words are similar.

$a_{13} = 10 - (a_1 + 3a_2 + \ldots + 3c_i + a_{i+1} + \ldots + 3a_{12}) \bmod 10,\ 2 \leq i \leq 12$ and

$a_{13} = 10 - (a_1 + 3a_2 + \ldots + 3d_i + a_{i+1} + \ldots + 3a_{12}) \bmod 10,\ 2 \leq i \leq 12$

suppose $3c_i \equiv k \pmod{10}$ and $3d_i \equiv w \pmod{10}$. Then

$a_{13} = 10 - (a_1 + 3a_2 + \ldots + k + a_{i+1} + \ldots + 3a_{12}) \bmod 10,\ 2 \leq i \leq 12$ and

$a_{13} = 10 - (a_1 + 3a_2 + \ldots + w + a_{i+1} + \ldots + 3a_{12}) \bmod 10,\ 2 \leq i \leq 12$.

In $\mathbb{Z}_{10}$, k = w since the code word is only made of digits between 0 and 9 hence there cannot be two different numbers between 0 and 9 which yield the same remainder modulo 10. Hence any single error on an even digit position will be detected.

Proposition 4
The ISBN-13 can detect any single error on any odd digit position.

Proof: By contradiction, suppose it cannot. This means that there exists two code words **x** and **y** which differ on one odd digit position say *i*. This position cannot be the check digit as shown earlier. Since *i* is odd, then *i*= 2n+1, n ∈ ℤ, 0 ≤ n ≤ 6. Let a_i and b_i be bit strings on **x** and **y** respectively which differ.

$a_{13} = 10 - (a_1 + 3a_2 + \ldots + a_i + 3a_{i+1} + \ldots + 3a_{12}) \bmod 10,\ 1 \leq i \leq 12$ and

$a_{13} = 10 - (a_1 + 3a_2 + \ldots + b_i + 3a_{i+1} + \ldots + 3a_{12}) \bmod 10,\ 1 \leq i \leq 12$

In $\mathbb{Z}_{10}$, $a_i = b_i$ since the code word is only made of digits between 0 and 9 hence there cannot be two different numbers between 0 and 9 which yield the same remainder modulo 10. Hence any single error on an odd digit position will be detected.

Corollary 3
The ISBN-13 can correct any single error on any digit position (even or odd).

Proof: Suppose an error has been detected on any digit position, a_i. To correct it, a digit is chosen such that equation 1.1.10.1 holds. Since computation is done modulo 10 and the code words are only made of bit strings between 0 and 9, there cannot be two digits between 0 and 9 satisfying the above equation. Thus the error detected can be corrected.

Proposition 5
In the ISBN-13, any single or multiple transposition error on even bit strings will go undetected.

Proof: In ISBN-13 computation of the check digit is done such that the equation below is satisfied

$a_{13} = 10 - (a_1 + 3a_2 + a_3 + 3a_4 + a_5 + 3a_6 + a_7 + 3a_8 + a_9 + 3a_{10} + a_{11} + 3a_{12}) \bmod 10$

The even bit strings are each multiplied by 3 then summed up modulo 10. If a single transposition error occurs such that an even bit string is transposed with another even bit string one in the same code word, the summation will not change and so the computation of a_{13} will not be affected.

Corollary 4
In the ISBN-13, any single or multiple transposition error on odd bit strings will go undetected.

Proof: The proof is as a consequence of proposition 5 above

Theorem 2.2.9
In the ISBN-13, double transposition errors may go undetected.

Proof: Suppose a double transposition error occurs on a code word C_1 to yield a code word C_2 such that for any two bit strings at digit positions a_x and a_y, the bit string at digit position a_x is transposed with the one at a_w whereas the bit string at digit position a_y is transposed with the one at a_p, in the same code word, where $x \neq w$ and $y \neq p$ and that $1 \leq x, y, w, p \leq 12$. If $x \neq w$, then there is no transposition! If $y \neq p$, then there is no transposition! Since the transposition occurs on the same code word, the received code word differs with the sent code word in four bit strings. Therefore the minimum distance $d(C_1, C_2) = 4$. Upon computation of the check digit for C_1 and C_2, the working is based on the parity of each of the bit string transposed so the check digit for C_2 will differ with that of C_1 if the transposed digit positions differ in their parity respectively. Otherwise if the digits transposed are of the same parity, the error will go undetected as shown in theorem 2.2.8 and Corollary 1 and 2 above.

The choices can be found by drawing a tree diagram involving the two parities to occupy the four positions. This yields to 2^4 options.

	a_x transposed to a_w		**a_y transposed to a_p**	
Cases	**a_x**	**a_w**	**a_y**	**a_p**
1	Odd	Even	Even	Even
2	Odd	Even	Even	Odd
3	Odd	Even	Odd	Even
4	Odd	Even	Odd	Odd
5	Odd	Odd	Even	Even
6	Odd	Odd	Even	Odd
7	Odd	Odd	Odd	Even
8	Odd	Odd	Odd	Odd
9	Even	Even	Odd	Even
10	Even	Even	Even	Odd
11	Even	Even	Odd	Odd
12	Even	Even	Even	Even
13	Even	Odd	Even	Even
14	Even	Odd	Even	Odd
15	Even	Odd	Odd	Even
16	Even	Odd	Odd	Odd

As far as parity of the transposed bit strings is concerned, Cases 5, 8, 11 and 12 represent multiple transpositions of bit strings with same parity and as proven earlier in (Kamaku et al., 2012)these errors cannot be detected.

In Cases 1, 4, 6, 7, 9, 10, 13 and 16, only one parity position is transposed. This represents a single transposition of bit strings with different parity. The error will not be detected if at these positions, the two weighted sums in the two codes are the same modulo 10 as discussed earlier in theorem 2.2.5. If not, then the error will be detected as follows:

Considering case 1:a_x (odd) transposed to a_w (even) whereas a_y (even) transposed to a_p (even)
For the sent code word: $a_x + 3a_w + 3a_y + 3a_p \equiv k \pmod{10}$, $k \in \mathbb{Z}$.
Received code word: $3a_x + a_w + 3a_y + 3a_p \equiv n \pmod{10}$, $n \in \mathbb{Z}$.

Since a_y and $a_p$have the same parity, they are both multiplied by 3 thus the overall sum is the same irrespective of the transposition. The difference in the two sums (sent and received code word) is therefore between $a_x + 3a_w$ and $3a_x + a_w$

Suppose $a_x + 3a_w \equiv (h \bmod 10)$ and $3a_x + a_w \equiv (y \bmod 10)$. This is the same as $(a_x + 3a_w) - h = 10c$ and $(3a_x + a_w) - y = 10d$ for some $c, d \in \mathbb{Z}$. Subtracting the second from the first yields $(2a_x - 2a_x) - (y - h) = 10(c - d)$. Hence $(2a_x - 2a_x) \equiv (y - h) \pmod{10}$

Suppose $y = h$, $y - h = 0$ then $2a_x - 2a_w \equiv 0 \pmod{10}$. Since computation is done in $\mathbb{Z}_{10}$, this can only happen when $a_i = a_w$ contradicting the fact that a transposition took place. Thus $y \neq h$ and so $a_x + 3a_w \neq 3a_x + a_w$and therefore $k \neq n$ hence the errors are detected. This shows that since the parity of a_x differs with that of a_w, if C_1 and C_2 have the same check digit, then an error(s) must have occurred. Cases 4, 6, 7, 9, 10, 13 and 16 follow a similar argument since in each, one of the transpositions occurs on digit positions with the same parity.

Cases 2, 3, 14, and 15 represent a double transposition of bit strings with different parity of the digit position. Similarly, the error will not be detected if at these positions, the two weighted sums in the two codes are the same modulo 10 as discussed earlier in corollary 1. If not, then the error will be detected as follows:

Consider case 14. ax (even) is transposed to aw(odd) whereas ay(even) is transposed to ap(odd)
Sent code word: $3ax + aw + 3ay + ap \equiv k \pmod{10}$, $k \in \mathbb{Z}$.
Received code word: $ax + 3aw + ay + 3ap \equiv n \pmod{10}$, $n \in \mathbb{Z}$.

If $k = n$, then it means the bit strings from the sent and received code words at these digit positions yield the same weighted sum modulo 10. Thus the error cannot be detected. If $k \neq n$, the errors are detected.

Conclusion and Recommendation

This paper analyses major weaknesses in both error detection and correction in the ISBN-13 code. It is shown that the code has major weaknesses both error detection and correction and hence cannot guarantee stability as far as this is concerned. This means that an error(s) in cataloguing of books in libraries could therefore face non-detectable errors. Improvement on error detection and correction is therefore inevitable to arrest the situation.

References

[1] Eric Weisstein, ISBN Code, http://mathworld.wolfram.com/ISBN.html (2010).
[2] E. Leo, The Coding of the ISBN, http://en.scientificcommons.org/leo-egghe (2010).

[3] Kamaku, P. W., Kivunge, B., & Wangeci, C. (2012). ON SOME PROPERTIES AND LIMITATIONS IN THE ISBN-13 CODE. International Electronic Journal of Pure and Applied Mathematics – IEJPAM , 4(3), 159–165.
[4] R. Ronald, E. Leo, On the detection of double errors in ISBN and ISSN-likecodes, http://en.scientificcommons.org/leo-egghe (2010).
[5] Viklund A., ISBN Information Home, http://isbn-information.com/index.html (2007).
[6] R.W. Hamming, Error detecting and error correcting codes, Bellsyst Technology Journal, 29 (1950), 147-160.

Author Profile

Waweru Kamaku, PhD, received the BSc. Mathematics and Computer Science(2006), MSc. in Pure Mathematics(2008) and PhD in Pure mathematics(2013) from the , Jomo Kenyatta University of Agriculture and Technology, Kenya. In 2015, he did a post-graduate diploma in Education and Technology at the University of Cape town South Africa where he researched more in learning with and through technology. He is now a lecturer in the Jomo Kenyatta University of Agriculture and Technology, Kenya.