

General rules of evaluating binary number divisibility on prime numbers

Alaa Ghazi Abdulbaqi¹, Ghadah A. Al-Sakkal², Yasir Hashim²

¹Department of Information Technology, Faculty of Applied Science, Tishk International University, Kurdistan Region, Erbil, Iraq

²Department of Computer Engineering, Faculty of Engineering, Tishk International University, Kurdistan Region, Erbil, Iraq

Article Info

Article history:

Received Jul 08, 2021

Revised Mar 29, 2022

Accepted Apr 06, 2022

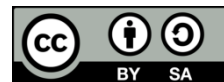
Keywords:

Binary
Divisibility
Evaluating
Number
Prime

ABSTRACT

This research paper is to define new rule to find the divisibility of stream of binary on any prime numbers 3, 5, 7 with reminder 0. In general, the divisibility of binary numbers is most important in many digital circuits and mathematical applications. This paper explains this new rule for evaluating the divisibility of any binary number on any prime number greater than 2. This rule will depend on separating the binary number into blocks of bits then processes each block separately in a special procedure to find the possible divisibility on the prime number. After testing this new rule with prime numbers 3, 5, and 7 as a sample of prime numbers, the finding shows that this rule provides fast and true results.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Yasir Hashim

Department of Computer Engineering, Faculty of Engineering

Tishk International University, Kurdistan Region, Erbil, Iraq

Email: yasir.hashim@tiu.edu.iq

1. INTRODUCTION

This research paper is to define new rule to find the divisibility of stream of binary on any prime numbers 3, 5, 7 ... with reminder 0. Cryptography theory relize on the principle of divisibility. In general, the divisibility of binary numbers is most important in many digital circuits and mathematical applications [1]-[3]. A positive integer P which is larger than 1 is called a prime number if $\forall n N, n | P \Rightarrow n = 1 \vee n = P$. A number which is not a prime is called a composite number. The fundamental theorem of arithmetic (FTA) says every integer larger than 1 can be expressed as a product of primes in a unique manner apart from their order. Prime numbers are used in cryptography to calculate the public and private keys. Its strength heavily depends on the difficulty of decomposing large integers into their factors. For instance, Diffie-Hellman used prime numbers in his key exchange [4]-[9].

Divisible codes were introduced for the first time by H. N. Ward in 1981 [10]-[15]. A divisible code is a linear code over a finite field whose code words all have weights divisible by some integer $\Delta > 1$, where Δ is called a divisor of the code. A binary linear code is said to be of (divisibility) level e if e is the greatest integer such that 2^e is a divisor of the code. The doubly-even binary self-dual codes may be viewed as level 2 divisible codes attaining the largest conceivable dimension for their lengths. Liu [16] gave an exact upper bound for the dimension of binary divisible codes in terms of code length and divisibility level (when the level is at least 3) and prove the uniqueness up to equivalence of the code attaining this bound, given the hypothesis that a certain non-zero weight exists. Also, this research paper proves that the hypothesis is true for level 3 divisible codes of maximum dimension with relatively short lengths.

2. METHODOLOGY

2.1. Mathematical background

The Fermat's little theorem [17]-[21] states that:

If p is any prime number and a is any integer such that p does not divide a , then:

$$a^{(P-1)} = 1 \pmod{P}$$

Also, it is possible to make use of the mod operation properties as: $a - cn = a \pmod{n}$, where a , c , and n are integers. So, the new formula will be: $a^{(P-1)} - 1 = 0 \pmod{P} = kP$, where k is an integer. Then if $a = 2$ is substituted, the relation will be as in (1). The methodology in this paper will assume that $P > 2$, so P is always an odd integer, and $(P-1)$ is always an even integer.

$$2^{(P-1)} - 1 = kP \quad (1)$$

2.2. Divisibility test by p by blocks

If it is assumed that the binary number B which its divisibility over P is under test has c bits $(b_{c-1}b_{c-2}b_{c-3}b_{c-4}b_{c-5}b_{c-6})_2$. So it can be represented as: $B = \sum_{j=0}^{c-1} b_j 2^j$. In the same time B can be re-written as series of m bits blocks (assuming c is a multiple of m by padding the most significant bits (MSBs) as 0),

$$B = \sum_{i=0}^{h-1} \sum_{r=0}^{m-1} b_{mi+r} 2^{mi+r} \quad (2)$$

Where $h = c/m =$ number of blocks in B , m is the minimum number which can satisfy the below (3)

$$2^m - 1 = kP \quad (3)$$

Where k is an integer and prime number $P > 2$. By comparing with (1), it can be concluded that m exists at least once and the maximum of m is: $Max(m) = (P-1)$. An algorithm to find the minimum value of m that can satisfy (3) can loop with increasing value of m as shown in Figure 1, Also we can express 2^m from (3) above as: $2^m = kP + 1$. If both sides have been raised to the power (i) then the result will be:

$$(2^m)^i = (kP + 1)^i \quad (4)$$

According to the binomial theorem formula [22]-[25]:

$$(a + b)^n = \sum_{k=0}^n \binom{n}{k} a^k b^{n-k}$$

If the binomial theorem is applied to the expression $(2^m)^i$. After substituting (4) then the result will be:

$$(2^m)^i = (kP + 1)^i = \sum_{k=0}^i \binom{i}{k} 1^{i-k} (kP)^k = 1 + kP \sum_{k=1}^i \binom{i}{k} (kP)^{k-1} \quad (5)$$

Since $\binom{i}{k}$ is always a positive integer, then the part $\left[kP \sum_{k=1}^i \binom{i}{k} a^k (kP)^{k-1} \right]$ is always divisible by P . Now the term (2^{mi}) can be taken as a common term out of the inner sum series in (2). So the result will be:

$$B = \sum_{i=0}^{h-1} 2^{mi} \sum_{r=0}^{m-1} b_{mi+r} 2^r$$

And by substituting (5) into it.

$$B = \sum_{i=0}^{h-1} \left(1 + kP \sum_{k=1}^i \binom{i}{k} (kP)^{k-1} \right) \sum_{r=0}^{m-1} b_{mi+r} 2^r$$

$$B = \sum_{i=0}^{h-1} [\sum_{r=0}^{m-1} b_{mi+r} 2^r] + kP \sum_{i=0}^{h-1} \left[\left(\sum_{k=1}^i \binom{i}{k} (kP)^{i-1} \right) \sum_{r=0}^{m-1} b_{mi+r} 2^r \right]$$

Let assume B_1

$$B_1 = \sum_{i=0}^{h-1} [\sum_{r=0}^{m-1} b_{mi+r} 2^r]$$

$$B_1 = \sum_{r=0}^{m-1} \left[\left(\sum_{i=0}^{h-1} b_{mi+r} \right) (2^r) \right] \tag{6}$$

Then B_1 became the divisibility test since the other part of B is a multiple of P . B_1 is a repetitive bitwise addition of the successive blocks of m bits. Now to complete the calculation of B_1 , each result of bitwise addition between blocks should be multiplied by its binary weight 2^r . From above discussion, we can conclude the below general theorem: to test the divisibility of binary number B on a prime number $P > 2$, then B can be separated into multiple blocks and bitwise additions of the blocks can be performed and if the resulted binary number is divisible by P then B is divisible by P .

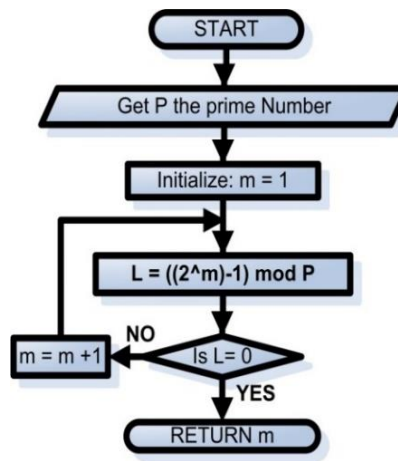


Figure 1. An algorithm to find the minimum block size m depending on P

2.2.1. Binary digit weight normalization

In order to further improve the divisibility test algorithm, the multiples of P from each bit weight 2^r should be removed. So the next step is to find the normalized bit weight matrix Q_r . Each entity Q_r in $Q[m]$ should not exceed the range shown in Figure 2, so it should satisfy the below inequality:

$$|Q_r| \leq \frac{(P-1)}{2}$$

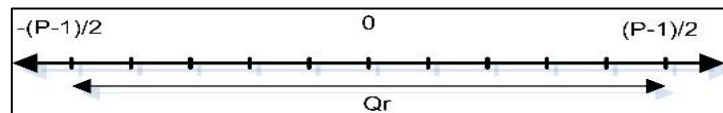


Figure 2. The range of normalized Q_r

Assuming that

$$Q_r = 2^r \text{ mod } P \text{ if } 0 \leq (2^r \text{ mod } P) \leq \frac{(P-1)}{2}$$

$$Q_r = (2^r \text{ mod } P) - P \text{ if } \frac{(P-1)}{2} < (2^r \text{ mod } P) \leq (P - 1)$$

Based on that, the following relation can be concluded

$$Q_r = 2^r - k_r P$$

Where k_r can be positive integer, negative integer, or zero, then

$$2^r = Q_r + k_r P \tag{7}$$

To construct the array

$$Q = [Q_0 \quad Q_1 \quad \dots \quad Q_{m-1}]$$

The algorithm shown in Figure 3 was used. The algorithm will generate $Q[m]$ for the given prime number input. To illustrate further the concept, Table 1 shows the block size m and $Q[m]$ for the prime numbers in the range $2 < P < 50$.

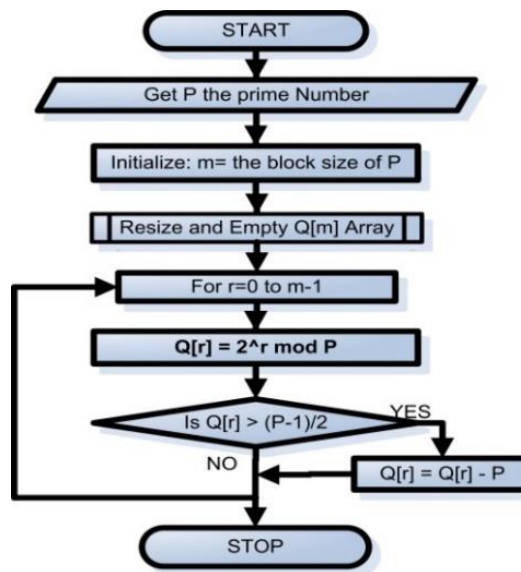


Figure 3. Construct Q and T matrices

Table 1. Listing the Q for all $2 < P < 50$

Prime (P)	Block size (m)	$Q[m]$
3	2	[1 -1]
5	4	[1 2 -1 -2]
7	3	[1 2 -3]
11	10	[1 2 4 -3 5 -1 -2 -3 -5]
13	12	[1 2 4 -5 3 6 -1 -2 -4 5 -3 -6]
17	8	[1 2 4 8 -1 -2 -4 -8]
19	18	[1 2 8 -3 -6 7 -5 9 -1 -2 -4 -8 3 6 -7 5 -9]
23	11	[1 2 4 8 -7 9 -5 -10 3 6 -11]
29	28	[1 2 4 8 -13 3 6 12 -5 -10 9 -11 7 14 -1 -2 -4 -8 13 -3 -6 -12 5 10 -9 11 -7 -14]
31	5	[1 2 4 8 -15]
37	36	[1 2 4 8 16 -5 -10 17 -3 -6 -12 13 -11 15 -7 -14 9 18 -1 -2 -4 -8 -16 5 10 -17 3 6 12 -13 11 -15 7 14 -9 -18]
41	20	[1 2 4 8 16 -9 -18 5 10 20 -1 -2 -4 -8 -16 9 18 -5 -10 -20]
43	14	[1 2 4 8 16 -11 21 -1 -2 -4 -8 -16 11 -21]
47	23	[1 2 4 8 16 -15 17 -13 21 -5 -10 -20 7 14 -19 9 18 -11 -22 3 6 12 -23]

By substituting the value of 2^r from (7) in (6), the below equation will be obtained:

$$B_1 = \sum_{r=0}^{m-1} [(\sum_{i=0}^{h-1} b_{mi+r}) (k_r P + Q_r)]$$

Which be processed further

$$B_1 = \sum_{r=0}^{m-1} [(k_r P \sum_{i=0}^{h-1} b_{mi+r} + Q_r \sum_{i=0}^{h-1} b_{mi+r})]$$

$$B_1 = P \sum_{r=0}^{m-1} [(k_r \sum_{i=0}^{h-1} b_{mi+r})] + \sum_{r=0}^{m-1} [(Q_r \sum_{i=0}^{h-1} b_{mi+r})]$$

Then B_2 will be

$$B_2 = \sum_{r=0}^{m-1} [(Q_r \sum_{i=0}^{h-1} b_{mi+r})] \quad (8)$$

Now if B_2 is defined as in (8). Then B_2 will be the part to decide about the divisibility of B since the other part of B_1 is multiple of P . In this case the algorithm should include the accumulation of each block of m bits in B into a temporary array called $S[m]$, $S = [S_0 \ S_1 \ \dots \ S_{m-1}]$ where:

$$S_r = \sum_{i=0}^{h-1} b_{mi+r} \quad (9)$$

Then the sum of each corresponding bits with the same weight can be taken to accumulate the matrix $S[m]$. Now substitute (9) into (8) we can (10). Then $total_sum$ is accumulated by multiplying each $Sum \ S_r$ by its corresponding weight Q_r similar to matrix multiplication. In order to convert this mathematical formula to a practical algorithm, if the resulting number is greater than the original prime number, i.e. if ($total \ sum > P$), then total sum is injected into the same algorithm in a repeating manner until a value of $total \ sum \leq P$ is reached, then divisibility can be judged by comparing the result to P or 0 as shown in Figure 4.

$$Total_Sum = B_2 = \sum_{r=0}^{m-1} [(Q_r \ S_r)] \quad (10)$$

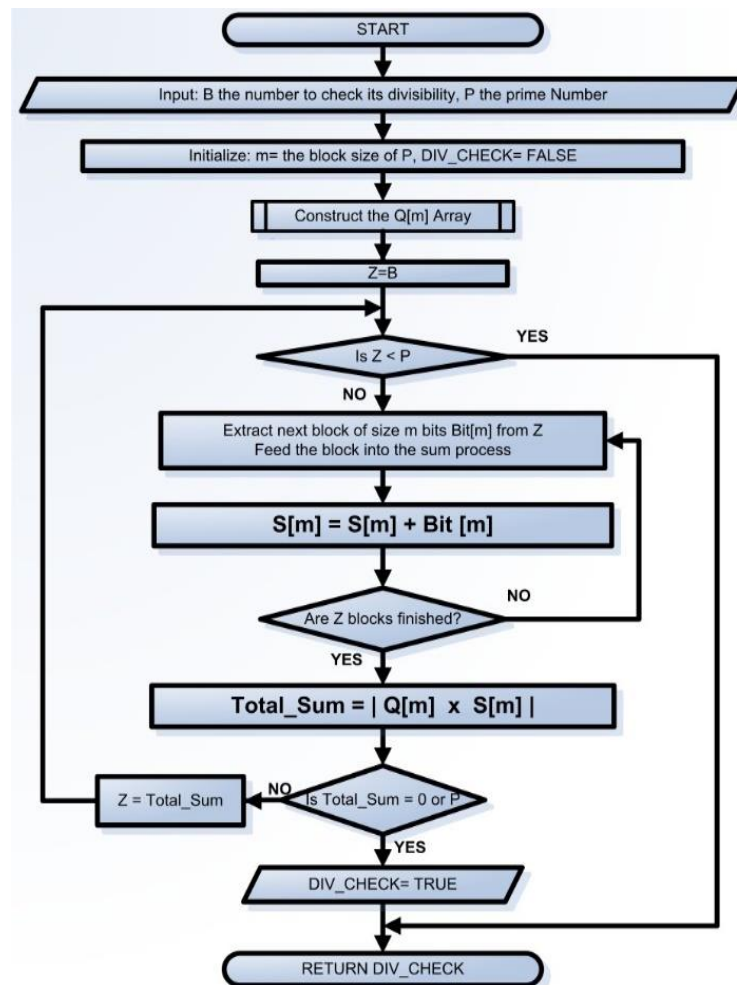


Figure 4. Testing the divisibility of number b on prime number P

2.3. Flowcharts of suggested system

In order to implement the general rules explained in this paper, below flow charts in Figure 1, Figure 2, and Figure 3 explain the procedures. Figure 1 explains the finding of block Size depending on P . Figure 2 illustrates how to construct Q array, and finally Figure 3 shows the algorithm for testing the divisibility of number b on prime number P .

2.4. Derving divisibility rules for first primes

In this section we are going to derive simple rules to find B_2 for the first prims in the natural numbers > 2 . The rules will be in terms of separating the binary number B . The binary number B will separate to small blocks and then manipulate each block by simple rule the adding or subtracting the results.

2.4.1. Divisibility by 3 rule

The algorithm can be specialized for $P = 3$ starting from (8). For $P = 3, m = 2, Q = [1 -1]$

$$B_2 = \sum_{r=0}^{m-1} [(Q_r \sum_{i=0}^{h-1} b_{mi+r})]$$

$$B_2 = \sum_{r=0}^1 [(Q_r \sum_{i=0}^{h-1} b_{2i+r})]$$

$$B_2 = (1)(b_0 + b_2 + \dots) + (-1)(b_1 + b_3 + \dots) = (b_0 + b_2 + \dots) - (b_1 + b_3 + \dots) \quad (11)$$

2.4.2. Divisibility by 5 rule

The algorithm can be specialized for $P = 5$ starting from (8). For $P = 5, m = 4, Q = [1 2 -1 -2]$

$$B_2 = \sum_{r=0}^{m-1} [(Q_r \sum_{i=0}^{h-1} b_{mi+r})]$$

$$B_2 = \sum_{r=0}^3 [(Q_r \sum_{i=0}^{h-1} b_{4i+r})]$$

$$B_2 = (1)(b_0 + b_4 + \dots) + (2)(b_1 + b_5 + \dots) + (-1)(b_2 + b_6 + \dots) + (-2)(b_3 + b_7 + \dots) \quad (12)$$

2.4.3. Divisibility by 7 rule

The algorithm can be specialized for $P = 7$ starting from (5), For $P = 7, m = 3, Q = [1 2 -3]$

$$B_2 = \sum_{r=0}^{m-1} [(Q_r \sum_{i=0}^{h-1} b_{mi+r})]$$

$$B_2 = \sum_{r=0}^2 [(Q_r \sum_{i=0}^{h-1} b_{3i+r})]$$

$$B_2 = (1)(b_0 + b_3 + \dots) + (2)(b_1 + b_4 + \dots) + (-3)(b_2 + b_5 + \dots) \quad (13)$$

3. TEST RESULTS AND DISCUSSION

3.1. Test the divisibility by 3

For $P = 3$ and $m = 2$ $Q = [1-1]$, as an application of (11) have been derived as in section 2.4.1, Table 2 shows two sample numbers to test their divisibility on 3. First the number is converted to binary and then it is segmented into blocks and then take the sum of each digit separately and finally multiply it by its normalized weight and check if the result is greater than 3 then take the result into another round and if the result is equal or less than 3 then if it is 0 or 3 then the number is divisible by 3 otherwise it is judged to be not divisible. Table 2 shows the test results for $P = 3$ sample no. 1, and also this table illustares the est results for $P = 3$ sample no. 2.

3.2. Test the divisibility by 5

For $P = 5, m = 4,$ and $Q = [1 2 -1 -2]$, as an application of (12) have been derived as in section 2.4.2, Table 3 shows two sample numbers to test their divisibility on 5. First the number is converted to binary and then it is segmented into blocks and then take the sum of each digit separately and finally multiply it by its normalized weight and check if the result is greater than 5 then take the result into another round and if the result is equal or less than 5 then if it is 0 or 5 then the number is divisible by 5 otherwise it is judged to be not divisible. Table 3 shows the test results for $P = 5$ sample no. 1, and also Table 3 explains the test results for $P = 5$ sample no. 2.

Table 2. Test results for $P = 3$ sample no. 1

Sample no	1	2
Decimal	569,841,236,988	5,698,426
Binary	100001001010110100101111001110111111100	010101101111001101111010
Round 1	10 00 01 00 10 10 11 01 00 10 11 11 00 11 00 11 10 11 11 11 00 13 10	01 01 01 10 11 11 00 11 01 11 10 10 7 8
	$Total_Sum = (-1) \times 13 + (1) \times 10 = -3$	$Total_Sum = (-1) \times + (1) \times 8 = -1$
Total sum	3	1
Divisible by 3	Yes	No

Table 3. Test results for $P = 5$ sample no. 1

Sample no	1	2
Decimal	569,841,236,990	5,698,426
Binary	100001001010110100101111001110111111110	0101011011111001101111010
Round 1	1 0 0 0 0 1 0 0 1 0 1 0 1 1 0 1 0 0 1 0 1 1 1 1 0 0 1 1 1 0 1 1 1 1 1 1 1 1 1 0 7 5 7 5 $Total_sum = (-2) \times 7 + (-1) \times 5 + (2) \times 7 + (1) \times 5 = 0$	0 1 0 1 0 1 1 0 1 1 1 1 0 0 1 1 0 1 1 1 1 0 1 0 2 4 5 4 $Total_sum = (-2) \times 2 + (-1) \times 4 + (2) \times 5 + (1) \times 4 = 6$
Round 2		0 1 1 0 0 1 1 0 $Total_sum = (-2) \times 0 + (-1) \times 1 + (2) \times 1 + (1) \times 0 = 1$
Total Sum	0	1
Divisible by 5	Yes	No

3.3. Test the divisibility by 7

For $P = 7$, $m = 3$, and $Q = [1\ 2\ -3]$, as an application of (13) have been derived as in section 2.4.3, Table 4 shows two sample numbers to test their divisibility on 7. First the number is converted to binary and then it is segmented into blocks and then take the sum of each digit separately and finally multiply it by its normalized weight and check if the result is greater than 7 then take the result into another round and if the result is equal or less than 7 then if it is 0 or 7 then the number is divisible by 7 otherwise it is judged to be not divisible. Table 4 shows the test results for $P = 7$ sample no. 1, and Table 4 illustrates the test results for $P = 7$ sample no. 2.

3.4. Test the divisibility by $2 < p < 1000$

A program in python v3 was developed based on the algorithms in Figure 1, Figure 2, and Figure 3. This program is to test the divisibility of the numbers from $2 < B < 10^{10}$ on the primes in the range $2 < P < 1000$. This python will compare its finding with results obtained with traditional mod operator and all results were compatible which proves the validity of the derived rules.

Table 4. Test results for $P = 7$ sample no. 1

Sample no	1	2
Decimal	569	5,698,427
Binary	001 000 111 001	010 101 101 111 001 101 111 011
Round 1	0 0 1 0 0 0 1 1 1 0 0 1 1 1 3 $Total_Sum = (-3) \times 1 + (2) \times 1 + (1) \times 3 = 2$	0 1 0 1 0 1 1 0 1 1 1 1 0 0 1 1 0 1 1 1 1 0 1 1 5 4 7 $Total_Sum = (-3) \times 5 + (2) \times 4 + (1) \times 7 = 0$
Total Sum	2	0
Divisible by 7	No	Yes

4. CONCLUSION

This research suggests a new rule for the divisibility of any binary number on any prime number greater than 2. This new rule depends on a special process on a separated blocks of the binary number with a special algorithm to test the possible divisibility on the prime number. After testing this new rule with prime numbers 3, 5 and 7 as a sample of prime numbers, the finding shows that this rule provides a fast and exact results. The following conclusions can be derived regarding the $Q(m)$ array for each prime number: i) the Q array represents a map for the corresponding prime divisibility; ii) the sum of the elements of Q array is always zero; iii) the block size is always $(P - 1)$ or one of its dividers; and iv) when the block size is $(P - 1)$, the second half of the array is the negative image of the first half, and the last element in the first half is exactly $(P - 1)/2$, so for example, $P = 11$, $m = 10$, $Q = [1\ 2\ 4\ -3\ 5\ -1\ -2\ -4\ 3\ -5]$.

ACKNOWLEDGEMENTS

The authors would like to thank Tishk International University (TIU) for their support.




REFERENCES

- [1] D. R. Lande, "Development of the Binary Number System and the Foundations of Computer Science," *The Mathematics Enthusiast*, vol. 11, no. 3, pp. 513-540, Dec. 2014, doi: 10.54870/1551-3440.1315.
- [2] A. Ahmad, "Investigation of some quite interesting divisibility situations in a signature analyzer implementation," *WSEAS Transactions on Circuits and Systems*, vol. 10, no. 9, pp. 299-308, Sep. 2011. [Online]. Available: https://www.researchgate.net/publication/256093039_Investigation_of_Some_Quite_Interesting_Divisibility_Situations_in_a_Sig_nature_Analyzer_Implementation.
- [3] T. Alexandrova, P. Boyvalenkov, and A. Dimitrov, "Binary (k, k)-Designs," *Mathematics*, vol. 8, no. 11, 2020, doi: 10.3390/math8111883
- [4] Y. Narayana and V. Yegnanarayanan, "On Prime number varieties and their applications," *Engineering and Applied Science Letters*, vol. 3, no. 3, pp. 30-36, Dec. 2020, doi: 10.30538/psrp-easl2020.0045.
- [5] H. M. Ahmed and R. W. Jassim, "Distributed Transform Encoder to Improve Diffie-Hellman Protocol for Big Message Security," *2020 3rd International Conference on Engineering Technology and its Applications (IICETA)*, 2020, pp. 84-88, doi: 10.1109/IICETA50496.2020.9318804.
- [6] P. Deshpande, S. Santhanalakshmi, P. Lakshmi, and A. Vishwa, "Experimental study of Diffie-Hellman key exchange algorithm on embedded devices," *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)*, 2017, pp. 2042-2047, doi: 10.1109/ICECDS.2017.8389808.
- [7] A. Taparia, S. K. Panigrahy, and S. K. Jena, "Secure key exchange using enhanced Diffie-Hellman protocol based on string comparison," *2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, 2017, pp. 722-726, doi: 10.1109/WiSPNET.2017.8299856.
- [8] T. K. Hazra, A. Mahato, A. Mandal, and A. K. Chakraborty, "A hybrid cryptosystem of image and text files using blowfish and Diffie-Hellman techniques," *2017 8th Annual Industrial Automation and Electromechanical Engineering Conference (IEMECON)*, 2017, pp. 137-141, doi: 10.1109/IEMECON.2017.8079577.
- [9] A. Ahmad and S. Ismail, "User Selective Encryption Method for Securing MANETs," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 8, no. 5, pp. 3103-3111, October 2018, doi: 10.11591/ijece.v8i5.pp3103-3111.
- [10] H. N. Ward, "Divisible codes," *Archiv der Mathematik*, vol. 36, pp. 485-499, 1981, doi: 10.1007/BF01223730.
- [11] S. Kurz, "No Projective 16-Divisible Binary Linear Code of Length 131 Exists," in *IEEE Communications Letters*, vol. 25, no. 1, pp. 38-40, Jan. 2021, doi: 10.1109/LCOMM.2020.3021939.
- [12] Y. M. Chee, G. Ge, and A. C. H. Ling, "Group Divisible Codes and Their Application in the Construction of Optimal Constant-Composition Codes of Weight Three," in *IEEE Transactions on Information Theory*, vol. 54, no. 8, pp. 3552-3564, Aug. 2008, doi: 10.1109/TIT.2008.926349.
- [13] M. Kiermaier and S. Kurz, "On the Lengths of Divisible Codes," in *IEEE Transactions on Information Theory*, vol. 66, no. 7, pp. 4051-4060, July 2020, doi: 10.1109/TIT.2020.2968832.




- [14] H. N. Ward, "A bound for divisible codes," in *IEEE Transactions on Information Theory*, vol. 38, no. 1, pp. 191-194, Jan. 1992, doi: 10.1109/18.108271.
- [15] X. Liu, "Weights Modulo a Prime Power in Divisible Codes and a Related Bound," in *IEEE Transactions on Information Theory*, vol. 52, no. 10, pp. 4455-4463, Oct. 2006, doi: 10.1109/TIT.2006.881708.
- [16] X. Liu, "Binary divisible codes of maximum dimension," *International Journal of Information and Coding Theory*, vol. 1, no. 4, pp. 355-370, Apr. 2010, doi: 10.1504/IJICOT.2010.032862.
- [17] S. Susanna, *Discrete Mathematics with Applications*, 4th edition, Boston, MA, USA: Cengage Learning Inc., 2020. [Online]. Available: https://www.academia.edu/42994708/Discrete_Mathematics_with_Application_by_Susanna_S_Epp
- [18] M. M. Wong, M. L. D. Wong, A. K. Nandi, and I. Hijazin, "AES S-box using Fermat's Little Theorem for the highly constrained embedded devices," *2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*, 2012, pp. 1039-1043.
- [19] G. Xiang and Z. Cui, "The Algebra Homomorphic Encryption Scheme Based on Fermat's Little Theorem," *2012 International Conference on Communication Systems and Network Technologies*, 2012, pp. 978-981, doi: 10.1109/CSNT.2012.208.
- [20] A. A. M Pushpa and S.Subramanian, "Study of Prime, Pseudoprime and applications of Pseudoprime," *Turkish Journal of Computer and Mathematics Education*, vol. 12, no. 9, pp. 934-939, Apr. 2021. [Online]. Available: <https://turcomat.org/index.php/turkbilmat/article/view/3333/2856>.
- [21] O. F. Florence, T. Ahmad, and A. A. Olalekan, "Dynamical system proof of Fermat's little theorem: An alternative approach," *Malaysian Journal of Fundamental and Applied Sciences*, vol. 14, no. 3, pp. 331-333, Sep. 2018, doi: 10.11113/mjfas.v14n3.1019.
- [22] B. Tatira, "Mathematics Education Students' Understanding of Binomial Series Expansion Based on the APOS Theory," *EURASIA Journal of Mathematics, Science and Technology Education*, vol. 17, no. 12, pp. 1-13, Oct. 2021, doi: 10.29333/ejmste/11287.
- [23] Y. Koh and S. Ree, "The Origin of Newton's Generalized Binomial Theorem," *Journal for History of Mathematics*, vol. 27, no. 2, pp. 127-138, Apr. 2014, doi: 10.14477/jhm.2014.27.2.127.
- [24] S. Aljohani, "History of Binomial Theory," *International Journal of Scientific & Engineering Research*, vol. 7, no. 4, pp. 161-162, 2016, <https://www.ijser.org/researchpaper/History-of-Binomial-Theory.pdf>
- [25] L. Misra, "A Review of Multiple Approaches for Binomial Theorem," *International Journal of Scientific and Research Publications*, vol. 8, no. 9, pp. 287-291, 2018, doi: 10.29322/IJSRP.8.10.2018.p8237.

BIOGRAPHIES OF AUTHORS






Alaa Ghazi Abdulbaqi    received the B.Sc. in Electronics and Communications Engineering from Al Nahrain University, Baghdad, Iraq in 1995. Then he received his MSc in Computer Engineering from Al Nahrain University, Baghdad, Iraq, in 1998. He is currently an Assistant Lecturer in the Department of Information Technology, Faculty of Science, Tishk International University, Erbil-Kurdistan, Iraq. He can be contacted at email: alaa.ghazi@tiu.edu.iq.



Ghadah A. Al-Sakkal    received the B.Sc. and master of Mathematics in Mathematical Science from Al-Mustansiriyya University of Baghdad, Iraq in 1987 and 1999 respectively. She is currently a Lecturer in the Department of Computer Engineering, Faculty of Engineering, Tishk International University, Erbil-Kurdistan, Iraq. She can be contacted at email: ghada.alsakkal@tiu.edu.iq.



Yasir Hashim    (SMIEEE) received the B.Sc. and master of Engineering in Electronics and Communications Engineering from the University of Mosul, Mosul, Iraq, in 1991 and 1995 respectively. He completed the Ph.D. in Electronics Engineering-Micro and Nanoelectronics from Universiti Science Malaysia (USM), Penang, Malaysia, in 2013. His research interests include Microelectronics and Nanoelectronic: Nanowire transistors, FinFET transistor, Multistage Logic Nano-inverters. He is currently a Senior Lecturer in the Department of Computer Engineering, Faculty of Engineering, Tishk International University, Erbil-Kurdistan, Iraq. He can be contacted at email: yasir.hashim@tiu.edu.iq, yasir.hashim@ieee.org.