IBIMA Publishing
Journal of Information Assurance & Cyber security
http://www.ibimapublishing.com/journals/JIACS/jiacs.html
Vol. 2016 (2016), Article ID 775081, 8 pages
DOI: 10.5171/2016.775081

Research Article

An Original Numerical Factorization Algorithm

George Kostopoulos

Information Assurance and Cyber security, University of Maryland University College, USA

Correspondence should be addressed to: George Kostopoulos;
george.kostopoulos@faculty.umuc.edu

Received date: 22 June 2016; Accepted date: 26 July 2016; Published date: 21 September 2016

Academic Editor: Furdu Iulian Marius

Copyright © 2016. George Kostopoulos. Distributed under Creative Commons CC-BY 4.0

Abstract

This paper presents a mathematical algorithm that supports the identification of prime numbers the product of which results in a given number. The associated research has found several dissimilar factorization algorithms. Considering that, none of the found algorithms resembles the one presented here; it is the author's belief that the presented algorithm is original. Using the algorithm presented in this paper, factorization is not a trial and error process, but a finite step by step process that can be very easily programmed and executed. Consequently, the ability to easily factorize a given number will disrupt existing numerical methodologies, while opening up possibilities for new cryptosyntesis and cryptanalysis areas of research and applications.

Keywords: Factorization; Numerical Analysis; Prime Numbers, cryptography

Introduction

In mathematics, factoring numbers into their set of prime values has been a major challenge, *". . . factoring numbers is a computationally difficult problem. It's easy for smaller numbers, but once you start dealing with very large numbers, it can take computers, days, months, years, even centuries to solve .There is no easy shortcut for factoring numbers – it's a trial and error process. "* [1].

In cryptography, researchers are *". . . mostly interested in the difficult RSA case, viz. n =pq with p; q primes of size approximately √n. This case is interesting, since if you can factor n then you can break the corresponding cryptosystem."* [2].

Apparently, there is a fear among researchers that *"As long as there is no general polynomial time algorithm for factoring large numbers, RSA may remain secure."* [3] The fears, however, are persistent because. *"Many security implementations in use are based on the difficulty for modern-day computers to perform large integer factorization*." [4].

Generally in factorization algorithms, one *". . would have to try all of the primes that are*

Cite this Article as: George Kostopoulos (2016), " An Original Numerical Factorization Algorithm", Journal of Information Assurance & Cyber security, Vol. 2016 (2016), Article ID 775081,
DOI: 10.5171/2016.775081

less than . . . (the number to be factorized, and until all of the primes are found which*) . . . when multiplied together . . .*" produce the number to be factorized [1].

In cryptography and in other numerical applications, factorization – because of its difficulty – has been a major asset. The development of this algorithm will force cryptographers to work harder, while opening new horizons in numerical analysis, especially in data compression.

Review of Related Literature

Considering the abundance of applications that employ the factorization of large integers, especially in cryptography, there are similarly numerous factorization methods. The most commonly known are briefly described in Table One. "*The methods . . . presented cannot, as yet, be analyzed formally . . .*" [2 p14]

Table 1: Most Known Factorization Methods

Algorithm Name	Comments
Pollard's rho	*". . . given $N \approx \sqrt{2p}$ random integers, we expect there to be a pair x, y which leads to a factorization of p. The question is how to find such a pair efficiently (testing all pairs will lead to a trivial O(p) algorithm)."* [2 p1]. The method involves a significant trial and error experimentation.
Pollard's p – 1	This algorithm is considered to be *". . . one of the fastest algorithms around, Lenstra's ECM, is based. The idea is to use Fermat's little theorem. . . "*. [2 p2].
Williams' p + 1	The method is limited by a number of parametric constraints, explained in the reference
Elliptic Curve	It is believed that this method *". . . turns out to be good enough in some cases . . ."* not supporting the factorization of any integer number. [2 p7].
Continued Fractions	For the factorization of large numbers *". . . we need to go through high precision calculations, which could also be costly."* [2 p8]
Quadratic Sieve	In the factorization mathematics *". . . the matrix we need to solve, in the linear algebra part, is sparse,"* posing a variety of problems. Furthermore, *". . . we note that there are also theoretic sub-cubic methods . . . but these aren't efficient in practice*." [2 p10]

A very thorough study on integer factorization algorithms concludes with the statement: "*There are no known algorithms which can factor arbitrary large integers efficiently*" [5]. With the factorization algorithm presented in this paper we may say that, now there is an algorithm that can factor arbitrary large integers efficiently.

The Developed Algorithm

The basic principle of the factorization algorithm, presented in this paper, is to treat the given integer, N, as the product of two numbers X and Y, where N=X*Y, or Y=N/X. The algorithm tracks the Y=X/N curve in unity steps, and stops when X*Y=N. The algorithm uses the square root of N as a starting point climbing upward along the curve.

Our aim in this climb is to reach either X*Y=N or X=1. In the first case, we stop and we evaluate each X and Y value separately to determine if either is a prime number. In the second case, we have successfully reached the end of the process having determined that Y is a prime number. Fig. 1 illustrates the general flowchart of the presented factorization algorithm.

In the climbing up on the Y=N/X curve, when the X&Y point falls in the X*Y<N area, Y is incremented Y in order for the X&Y point to get closer to the curve, when the X&Y point falls in the X*Y>N area, X is decremented to get closer to the curve.

The presented algorithm is an iterative process using a step by step approach. Each iteration produces two factors, identified as X and Y. If X*Y=P, we stop. If X=1, then Y=N and N is a prime number. If X<>1, no conclusion is drawn, and the X and Y values are individually analyzed. It should be noted that, at the starting point of the iteration, the values of the X&Y pair are the integers most close to R=sqrt(N), the value of the square root of N, where X<R and Y>R. Should R be an integer itself, then X=Y and X*Y=N, with N being a perfect square. In this case, R is being evaluated to determine if in itself is a prime number. From this initial point on, the value of X is steadily decremented and that of Y is steadily incremented. The X*Y product is continuously tested against the value of N, and when X*Y=N and the X value reaches unity, then we conclude that the corresponding Y is a prime number.

An Example

In this section, there is an example demonstrating the use of the above factorization algorithm and the logic behind it. Here, N is assumed to be 19. Table Two lists the coordinates of the points, as we climb around curve X*Y=N.

Fig. 2 shows the path of the X&Y pair ascending along the N=19=X*Y or Y=19/X curve, where the square root of N is R=4.35889...., identified as *Point Zero*. The starting point, or *Point One,* is the X&Y integral pair nearest to the X=Y=sqrt(N)=R point, where X<R and Y>R. Thus, in this case, the starting coordinates become X<R or X=4 and Y>R or Y=5.

At *Point One*, P=X*Y=4*5=20. Because the point falls in the X*Y>N (right hand side) area, the next step is to decrement X creating *Point Two*. The zig-zag continues ending up in one of two options.. One is the case where N is a prime number, as in Fig. 1, where N=19. The other case is where product X*Y=N but X<>1, and we need to analyze X and Y individually.

George Kostopoulos (2016), Journal of Information Assurance & Cybersecurity, DOI: 10.5171/2016.775081

If in the current step product **X*Y is than N**, in the following step the value of X is decremented. If in the current step product **X*Y is less than N**, in the following step the value of Y is incremented. When **X*Y reaches N,** the process stops. If **X=1,** as it is in this case, **Y** is a prime number. If **X>1, N** is the product of **X** and **Y,** and **X** and **Y** are individually assessed to determine their prime status.

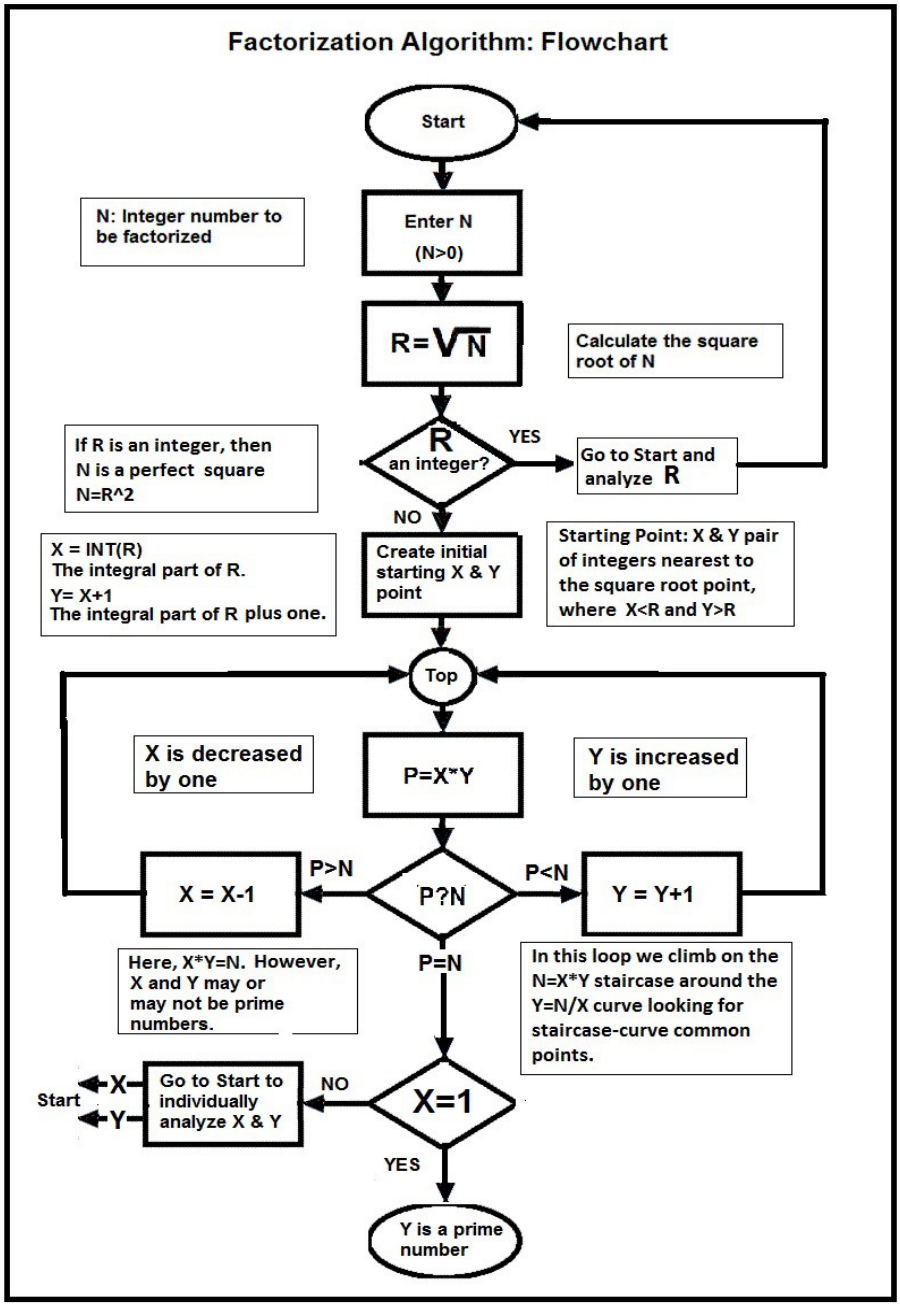

Fig. 1. Flowchart of the Factorization Algorithm.

Conclusion

The development of a simple and reliable algorithm for the factorization of any integer, as presented here, creates new research opportunities in numerical analysis, cryptography and in data compression and data representation. Factorization is the backbone of most cryptographic protocols. The availability of the presented factorization algorithm bases cryptography less on computing power, and more on cryptographic ingenuity.

Table 2: Steps leading to the determination that number 19 is a Prime Number

Point	X	Y	P=X*Y	P ? N	Next?
Zero	**4.358...**	**4.358...**	**19**	**P = N**	
One	**4**	**5**	**20**	**P > N**	**X-**
Two	**3**	**5**	**15**	**P < N**	**Y+**
Three	**3**	**6**	**18**	**P < N**	**Y+**
Four	**3**	**7**	**21**	**P > N**	**X-**
Five	**2**	**7**	**14**	**P < N**	**Y+**
Six	**2**	**8**	**16**	**P < N**	**Y+**
Seven	**2**	**9**	**18**	**P < N**	**Y+**
Eight	**2**	**10**	**20**	**P > N**	**X-**
Nine	**1**	**10**	**10**	**P < N**	**Y+**
Ten	**1**	**11**	**11**	**P < N**	**Y+**
Eleven	**1**	**12**	**12**	**P < N**	**Y+**
Twelve	**1**	**13**	**13**	**P < N**	**Y+**
Thirteen	**1**	**14**	**14**	**P < N**	**Y+**
Fourteen	**1**	**15**	**15**	**P < N**	**Y+**
Fifteen	**1**	**16**	**16**	**P < N**	**Y+**
Sixteen	**1**	**17**	**17**	**P < N**	**Y+**
Seventeen	**1**	**18**	**18**	**P < N**	**Y+**
Eighteen	**X = 1**	**19**	**19**	**P = N**	

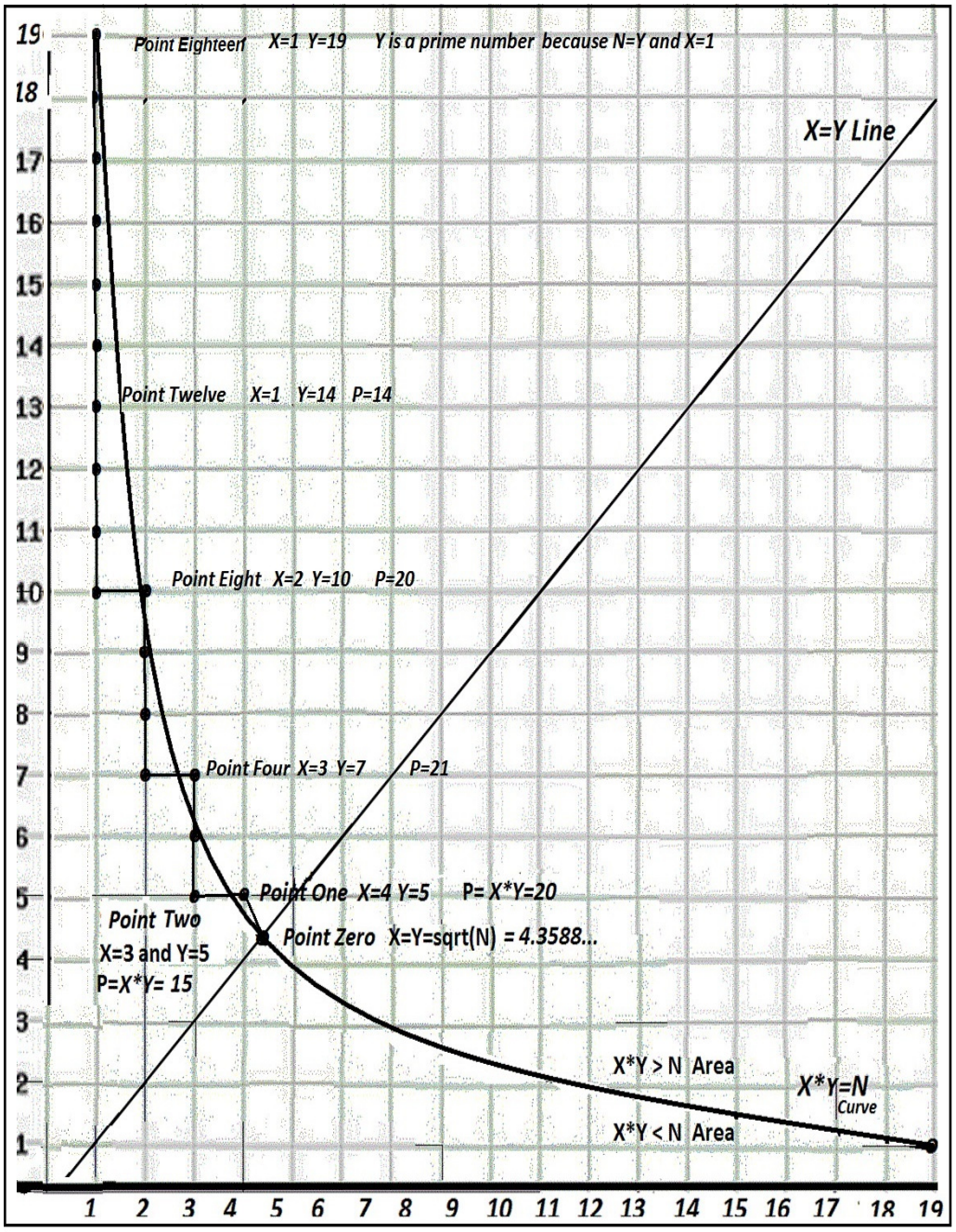

Fig. 2 Demonstration of the Factorization Algorithm identifying N=19 as a prime number.

References

1. Learn Cryptography.com http://learncryptography.com/prime-factorization/

2. Yuval Film, Factorization Methods: Very Quick Overview by http://www.cs.toronto.edu/~yuvalf/Factorization.pdf

3. Kevin Chu, RSA Cryptography: Factorization by http://wstein.org/edu/2010/414/projects/chu.pdf

4. Travis L. Swaim, Quantum Computing and Cryptography Today: Preparing for a Breakdown http://essic.umd.edu/joom2/index.php/faculty-and-staff?layout=user&user_id=171&dir=JSROOT%2Ftswaim1&download_file=JSROOT%2Ftswaim1%2FQuantum+Computing+and+ Cryptography+Today.pdf

5. Connelly Barnes (2004) Integer Factorization Algorithms http://connellybarnes.com/documents/factoring.pdf

George Kostopoulos (2016), Journal of Information Assurance & Cybersecurity, DOI: 10.5171/2016.775081

APPENDIX : A Selected Bibliography on Numbers Factorization

RSA Cryptography: Factorization	http://wstein.org/edu/2010/414/projects/chu.pdf
RSA Challenge	https://en.wikipedia.org/wiki/
RSA Factorization modulus	https://eprint.iacr.org/2010/006.pdf
Modern Factoring Algorithms	http://www.cs.columbia.edu/~rjaiswal/factoring-survey.pdf
Deterministic Algorithms	http://arxiv.org/ftp/arxiv/papers/1308/1308.2891.pdf
Factorization of a Large Number http://www.cse.buffalo.edu/faculty/miller/Courses/CSE710/Factorization_of_a_Large_number.pdf	
Online Factorization Tools http://www.virtuescience.com/prime-factor-calculator.html http://www.calculatorsoup.com/calculators/math/prime-factors.php http://www.numberempire.com/factoringcalculator.php	
Factorization methods	
Fermat's	https://en.wikipedia.org/wiki/Fermat%27s_factorization_method
Euler's	https://en.wikipedia.org/wiki/Euler%27s_factorization_method
Cholesky	https://en.wikipedia.org/wiki/Cholesky_decomposition
Cryptography. UAM 2010-2011 Factorization algorithms	http://www.uam.es/personal_pdi/ciencias/fchamizo/asignaturas/cripto1011/factorization.pdf