Advances in composite integer factorization

Aldrin W. Wanambisi[1*] Shem Aywa[2], Cleophas Maende[3], Geoffrey Muchiri Muketha[4]

1. School of Pure and Applied Science, Mount Kenya University, P.O box 342-00100, Thika, Kenya.
2. Dept of Mathematics, Masinde Muliro University of Science and Technology, P.O Box 150-50100, Kakamega, Kenya.
3. School of Post graduate studies, Mount Kenya University, P.O box 342-00100, Thika, Kenya.
4. Dept of Computer Science, Masinde Muliro University of Science and Technology, P.O Box 150-50100, Kakamega, Kenya.

* E-mail of the corresponding author: wawanambisi@gmail.com

Keywords: *Integer factorization, Prime number, logarithm.*

Abstract

In this research we propose a new method of integer factorization. Prime numbers are the building blocks of arithmetic. At the moment there are no efficient methods (algorithms) known that will determine whether a given integer is prime or and its prime factors [1]. This fact is the basis behind many of the cryptosystems currently in use.

1.0 Introduction

There are no known algorithms which can factor arbitrary large integers efficiently. Probabilistic algorithms such as the Pollard rho and Pollard *p*-1 algorithm are in most cases more efficient than the trial division and Fermat factorization algorithms. However, probabilistic algorithms can fail when given certain prime products: for example, Pollard's rho algorithm fails for $N = 21$ [6]. Integer factorization algorithms are an important subject in mathematics, both for complexity theory, and for practical purposes such as data security on computers [3].

2.0 Basic Concepts

An integer $p \geq 2$ is prime if it has no positive divisors other than 1 and itself. An integer greater than or equal to 2 that is not prime is composite. An integer $n \geq 2$ is composite if and only if it has factors a and b such that $1 < a < n$ and $1 < b < n$. If n > 1 then there is a prime p such that $p \mid n$ where $p \mid n$ denotes p divides n [8].

2.1 Prime factorization algorithms

Many algorithms have been devised for determining the prime factors of a given number (a process called prime factorization). They vary quite a bit in sophistication and complexity [1], [2]. It is very difficult to build a general-purpose algorithm for this computationally "hard" problem, so any additional information that is known about the number in question or its factors can often be used to save a large amount of time.

The simplest method of finding factors is so-called "direct search factorization" (a.k.a. trial division). In this method, all possible factors are systematically tested using trial division to see if they actually divide the given number. It is practical only for very small numbers.

The fastest-known fully proven deterministic algorithm is the Pollard-Strassen method [6].

2.2 Integer factorization

The factorization of a number into its constituent primes, also called prime decomposition. Given a positive integer $n \geq 2$, the prime factorization is written

$$n = p_1^{\alpha_1} p_2^{\alpha_2} \cdots p_k^{\alpha_k},$$

where the p_i s are the k prime factors, each of order α_i. Each factor $p_i^{\alpha_i}$ is called a primary. Prime factorization can be performed in *Mathematica* using the command Factor Integer[n], which returns a list of (p_i, α_i)pairs.

Through his invention of the Pratt certificate, Pratt (1975) [4] became the first to establish that prime factorization lies in the complexity class NP.

The number of *digits* in the prime factorization of n = 1, 2, ..., are 1, 1, 1, 2, 1, 2, 1, 2, 2, 2, 2, 3, (Sloane's A050252). [4]

In general, integer factorization is a difficult problem, and many sophisticated integer factorization algorithms have been devised for special types of numbers.

Interestingly, prime numbers p equal to 1 (mod 4) can always by factored into Gaussian primes in the form

$$p = -i\,(R + I\,i)\,(I + R\,i),$$

where the real and imaginary parts are inverted in the two parts, while prime numbers equal to 3 (mod 4) cannot be factored into Gaussian primes. This is directly related to Fermat's $4n+1$ theorem.

2.3 Examples of factorization algorithms

2.3.1 Trial Division

A brute- force method of finding a divisor of an integer n by simply plugging in one or a set of integers and seeing if they divide n. Repeated application of trial division to obtain the complete prime factorization of a number is called

direct search factorization. An individual integer being tested is called a trial divisor. [1]

2.3.2 Direct search factorization

Direct search factorization is the simplest (and most simple-minded) prime factorization algorithm. It consists of searching for factors of a number by systematically performing trial divisions usually using a sequence of increasing numbers. Multiples of small primes are commonly excluded to reduce the number of trial divisors, but just including them is sometimes faster than the time required to exclude them. Direct search factorization is very inefficient, and can be used only with fairly small numbers.

When using this method on a number n, only divisors up to $\lfloor n \rfloor$ (where $\lfloor x \rfloor$ is the floor function) need to be tested. This is true since if all integers less than this had been tried, then

$$\frac{n}{\lfloor \sqrt{n} \rfloor + 1} < \sqrt{n} \qquad (1)$$

In other words, all possible factors have had their cofactors already tested. It is also true that, when the smallest prime factor p of n is $> \sqrt[3]{n}$, then its cofactor m (such that $n = pm$) must be prime. To prove this, suppose that the smallest p is $> \sqrt[3]{n}$. If $m = ab$, then the smallest value a and b could assume is p. But then

$$n = pm = pab \geq p^3 > n \qquad (2)\ ,$$

which cannot be true. Therefore, m must be prime, so $n = p_1 p_2$ [1].

3.0 Results

3.1 Introduction

In this section, we focus on the aspect of integer factorization. The so far proposed algorithms have proved not to be efficient namely trial method, the direct search method, the fermat factorization method and GNFS [2]. These algorithms do not run polynomial times.

3.3 The proposed algorithm

The fundamental theorem of arithmetic states that every positive integer can be written uniquely as a product of primes, when the primes in the product are written in non decreasing order. The fundamental theorem of arithmetic implies that any composite integer can be factored. Let n be a composite integer a product of two primes p and q which are not necessary equal but close as is in the RSA [3]. Now clearly the logarithm of the two primes is approximately ½ (log n). After the approximate logarithm has been obtained the nearest prime can be determined through direct search. For example 21, first 1/2 (log 21) = 07563, the nearest primes with this 3 and 7. For Blum integers, which has extensively been used in the domain of cryptography, are integers with form $p^{k_1} q^{k_2}$, where p and q are different primes both ≡ 3 mod 4 and $k_1 and\ k_2$ are odd integers. These integers can be divided two types:

1. $M = pq$, hence $log\ p \approx {}^1/_2\ (\log M) \approx \log q$, the actual values of p and q can be estimated from primes nearest to the integer equivalent to $\sqrt{M}$.

2. $M = p^{k_1} p^{k_2}$, where at least one of $k_1 and\ k_2$ is greater than 1, hence $k_1 \log p \approx {}^1/_2\ (\log M) \approx k_2 \log p$ similarly the ${p_1}^{k_1}$ and ${p_2}^{k_2}$ can be estimated as $\sqrt{M}$.

This estimation algorithm reduces the number of steps that can be used determine the prime factors of composite integers. The table below shows some composite integers and the prime factors based on the estimation algorithm:

Table 3.0

Integer	**½ (log *n*) (4 d.p)**	***p***	***q***
21	0.7563	3	7
568507	2.8774	751	757
7064963	6.8491	2657	2659
31945104	7.5044	5651	5653

4.0 Conclusion

The above algorithm can be used to factor large integer with relatively better efficiency compared to the existing algorithms. Though it could be argued the algorithm is more like direct search but a number of steps are significantly reduced. I believe that a study in trends in differences between consecutive primes is the way to go in seeking a much faster algorithm to factor composite integers. A closer look at differences between primes reveals the different

categories of primes for example twin primes, primes $\equiv 3 \pmod 4$ have their differences as 2, 4 etc. If these differences can be linked to the products obtained when such primes are multiplied then the number of steps required to factorize such will be greatly reduced. The differences between any two primes which are close but not necessarily equal are always in the units place

Authors' contributions

All authors contributed to the conceptualisation of the paper. Wanambisi A.W. did the initial review, the selection of abstracts, and the identification of papers to be included in the final review. All authors contributed to the assessment of papers. All authors reviewed the results of the analysis. Wanambisi drafted the manuscript, and all authors contributed to its completion.

Acknowledgements

Thanks to those who have been instrumental in the success of this research: The Masinde Muliro University of Science and Technology, the adviser, for participating in this research study and for their support of this study

References

[1]. Connelly B. "*Integer Factorization Algorithms"*. December 7, 2004
[2]. "*General number field sieve*." From Wikipedia, an online encyclopedia. November 13, 2004. Available: http://en.wikipedia.org/wiki/GNFS
[3]. Wesstein, Eric W. "*RSA Encryption*." From Mathworld, an online encyclopedia. April, 2001. Available: http://mathworld.wolfram.com/RSAEncryption.html
[4]. "*Integer factorization* . Difficulty and complexity." From Wikipedia, an online encyclopedia. October 30, 2004. Available: http://en.wikipedia.org/wiki/Integer_factorization
[5]. Weisstein, Eric W. "Fermat, Pierre de." From MathWorld, an online encyclopedia. Available: http://scienceworld.wolfram.com/biography/Fermat.html
[6]. Weisstein, Eric W. "*Pollard Rho Factorization*." From MathWorld, an online encyclopedia. December 28, 2002. Available: http://mathworld.wolfram.com/PollardRhoFactorizationMethod.html
[7]. Weisstein, Eric W. "*Brent's Factorization Method*." From MathWorld, an online encyclopedia. December 28, 2002. Available: http://mathworld.wolfram.com/BrentsFactorizationMethod.html
[8] Hefferon J. *Elementary Number Theory*, December, 2003