International Journal of Mathematics and Statistics Studies

Vol.1, No.1, March 2013, pp.16-20

Published by European Centre for Research, Training and Development, UK (www.ea-journals.org)

ALGEBRAIC APPROACH TO COMPOSITE INTEGER FACTORIZATION

Aldrin W. Wanambisi[1*]

School of Pure and Applied Science, Mount Kenya University, P.O box 553-50100, Kakamega, Kenya.

Shem Aywa[2]

Department of Mathematics, Masinde Muliro University of Science and Technology, P.O Box 190-50100, Kakamega, Kenya.

Cleophas Maende[3]

School of Post Graduate Studies, Mount Kenya University, P.O box 342-00100, Thika, Kenya.

Geoffrey Muchiri Muketha[4]

Department of Information Technology, Meru University of Science and Technology, P.O Box 672 - 60200, Meru, Kenya.

Abstract: *There various algorithms that can factor large integers but very few of these algorithms run in polynomial time. This fact makes them inefficient. The apparent difficulty of factoring large integers is the basis of some modern cryptographic algorithms. In this paper we propose an algebraic approach to factoring composite integer. This approach reduces the number of steps to a finite number of possible differences between two primes.*

Keywords: Composite integer, Primes, Algebraic decomposition, Algorithm

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 [6]. However, probabilistic algorithms can fail when given certain prime products: for example, Pollard's rho algorithm fails for $N = 21$. Integer factorization algorithms are an important subject in mathematics, both for complexity theory, and for practical purposes such as data security on computers [3]. From [5], the differences between consecutive primes provide an important base in developing new approaches in integer factorization.

2.0 Basic concepts

2.1 Prime Numbers

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 a prime is composite.

2.2 Lemma

An integer $n \geq 2$ is composite if and only if it has factors a and b such that $1 < a < n$ and $1 < b < n$.

Proof

Let $n \geq 2$ The 'if' direction is obvious. For 'only if', assume that n is composite. Then it has a positive integer factor a such that $a \neq 1$ and $a \neq n$. This means that there is a b with $n = ab$. Since n and a are positive, so is b. Hence $1 \leq a$ and $1 \leq b$. By Divisibility properties of integers, $a \leq n$ and $b \leq n$. Since $a \neq 1$ and $a \neq n$ we have

$1 < a < n$. If b = 1 then a = n, which is not possible, so $b \neq 1$. If b = n then a = 1, which is also not possible. So $1 < b < n$, finishing this half of the argument [6].

2.3 Lemma

If $n > 1$, then there is a prime p such that $p \mid n$.

Proof

Let S denote the set of all integers greater than 1 that have no prime divisor. We must show that S is empty. If S is not empty then by the Well-Ordering Property it has a smallest member; call it m. Now m > 1 and has no prime divisor. Then m cannot be prime (as every number is a divisor of itself). Hence m is composite. Therefore (by result above) m = ab where $1 < a < m$ and $1 < b < m$. Since1 $< a < m$, the factor a is not a member of S. So a must have a prime divisor p. Then $p \mid a$ and $a \mid m$, so by divisibility properties of integers, $p \mid m$. This contradicts the assumption that m has no prime divisor. So the set S must be empty [6].

3.0 Existing Integer factorization algorithms

3.1 Trial Division

Trial division is 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].

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

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

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

3.3 The logarithm estimation 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 [5].

4.0 The Proposed Algebraic Approach

This section discusses the results of the research. A study of differences of consecutive primes reveals trends that we exploit in this paper.

4.1 Twin primes

Twin primes have a difference of 2. Thus if p and q are any two consecutive twin primes then the product pqcan be given by $(n-1)(n+1)$. Now consider a composite integer $m = pq$ that is a product of twin primes. Then

$$(n-1)(n+1) = m$$

$$n^2 + 1 = m$$

$$n = \pm\sqrt{m+1}$$

Taking the appropriate value of *n* to be approximately *p* then *q* follows at once.

4.2 Blum integers

A Blum integer is a composite integer that is a product of two primes both congruent to 1 modulo 4. The difference between any two such consecutive primes is 4. Thus if *p* and *q* are two primes both congruent to 1 modulo 4, then

$$|p-q| = 4$$

Therefore if we let *p* to be say *n*-2 and *q* to be say *n*+2, then a Blum integer $m = pq$

$$(n-2)(n+2) = m$$

$$n^2 - 4 = m$$

$$n^2 = m + 4$$

$$n = \pm\sqrt{m+4}$$

An appropriate value of *n* gives us approximate p and q hence the prime factors of our composite integer.

4.3 Other composite integers

In this section we consider other composite integers which are products of primes with differences of 6, 8, 10, 12, 14, 16 e.t.c. now just like the case of Twin and Blum integers, if $m = pq$ is any composite integer, with difference of say 6, then clearly

$$|p - q| = 6$$

Thus if we let $p = n - 3$ and $q = n + 3$ then a composite integer $m = pq$

$$(n-3)(n+3) = m$$

$$n^2 - 9 = m$$

$$n^2 = m + 9$$

Therefore $n = \pm\sqrt{m+9}$

An appropriate value of *n* gives approximate values of the prime factors of the composite integer.

For any general composite integer $m = pq,$ with the difference $|p - q|$ the prime factors are approximately

$$n = \pm\sqrt{m + \left(\frac{|p-q|}{2}\right)^2} \qquad (3)$$

Using the relation (3), we can obtain prime factors of composite integers on condition that the integer is a product of two primes no matter how large. This method reduces the steps that lead to factorization of a large integer to say polynomial time. If for example we take the case of RSA cryptosystem which is based on the prime factorization problem in which the primes are relatively close, the steps taken to arrive at the prime factors are greatly reduced. This then can be done in polynomial time.

5.0 Results

A comparison of the approach above with other algorithms like the trial method, the direct search, the General Number Field Sieves [2], the logarithm method shows a great reduction in the number of steps. The algebraic approach uses differences, which repeat themselves across the set of primes. Considering any two consecutive primes which are sufficiently large, we find that the difference ranges approximately between 2 and 24**.** Working with this approach makes the number of steps to arrive at the prime factors to the number of even numbers in this range. Then the question is how large a composite integer can be to be efficiently decomposed by the algorithm? If this approach is combined by the logarithm then any integer no matter how large can be factored though with some bit of approximation.

6.0 Conclusion

The algebraic approach brings out important revelations that can be exploited in the quest of efficiently factoring large composite integers. Though this puts the security of the world's money and many government secrets at risk, it's an important move in the growth of knowledge. The algorithm runs in polynomial time.

If a study can be carried to establish the maximum difference between two consecutive primes and their relationships, this will strengthen the approach. A cryptographic primitive based on these differences is also likely to be probabilistic and very efficient and commercially viable comparable to the RSA and ECC.

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] Wanambisi A.W., Aywa S., Maende C. Muketha G. M., *Advances in composite integer factorization,* Journal of Mathematical Theory and Modeling, March 2013

[6] Hefferon J. *Elementary Number Theory*, December, 2003

Authors' contributions

All authors contributed to the conceptualisation of the paper which is an excerpt of W.A.W. PhD dissertation. W.A.W. did the initial review, the selection of abstracts, and the identification of papers to be included in the final review. SA, CM and GMM contributed to the assessment of papers and reviewed the results of the analysis. W.A.W 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.

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