International Journal of Electronics and Applied Research (IJEAR) **2015, vol. 2, issue 1**
Published Online June 2015 issue (*http://eses.co.in/ESES Journal)* ISSN NO: 23950064

Finding Factors of a Number Using Steepest Ascent Hill Climbing

B. Choudhury[1*] and S. Neog[1]

[1]*Department of Information Technology, North Eastern Hill University, Shillong*
**bhargabchoudhury24@gmail.com*

Abstract

Breaking a large integer number into its factors is a famous problem in the field of Mathematics and Cryptography. This problem is called as Integer Factoring Problem (IFP). For given p and q, it is easy to compute N=p×q. But for a given N it is difficult to compute p and q. It is a well known NP problem and security of many cryptosystem is based on difficulty of integer factorization. One of the most well-known public key cryptosystem is Rivest-Shamir-Adleman (RSA) .Security of RSA cryptosystem is based on IFP. This paper is an attempt to find factors of an integer number using artificial intelligence technique called steepest ascent hill climbing. Fermat factorization rewrites a composite number N as the difference of squares, $N=X^2-Y^2$ and the heuristic function used in the process of finding factors is derived from this formula. The experimental results validate that the procedure proposed is successful in finding the factors.

Key word: Integer; Integer; Factoring Problem; RSA.

1. Introduction

In number theory, integer factorization or prime factorization is the decomposition of a composite number into smaller non-trivial divisors, which when multiplied together equals to the original integer. Many cryptographic protocols are based on the difficulty of factoring large composite integers or a related problem, for example: RSA. The RSA cryptosystem proposed in 1978 by Rivest, Shamir and Adlenman is the most well known public key cryptosystem [Rivest et al., 1978]. It is widely used to secure the information in the insecure channel. It is also implemented in most Web servers and browsers, and in most commercially available security products. RSA algorithm uses a pair of keys to encrypt and decrypt a message: one key is used to encrypt, called public key; and the other key is used to decrypt, called private key. The security of RSA is based on the difficulty of integer factorization. The integer factorization

problem is a well-known topic of research within both academia and industry. It consists of finding the prime factors for any given integer number.

This paper describes hill climbing as an artificial intelligence technique to find the factors of an integer number.

2. RSA Cryptosystem

The most common public key algorithm is RSA Cryptosystem. The mathematical structure of the RSA function is quite simple. It is based on basic algebraic operations on large integers. RSA uses two exponent e and d, where e is public and d is private. RSA algorithm is described below:

2.1 Key Generation

Bob (a pseudo name) uses the following steps to create his public key and private key :

1: Select two large prime numbers p and q such that p and q is not equal
2: n = p * q
3: $\Phi(n) = (p*1)(q*1)$
4: Select e such that $1 < e < \Phi(n)$
5: $d \equiv e^{-1} \bmod \Phi(n)$
6: Public key: (e; n) and Private key: d

After key generation, Bob announce ‘e, n’ as public key and keeps ‘d’ as private key.

2.2 Encryption

Anyone can send a message to Bob using his public key. The size of the plaintext P must be less than n, which means that if the size of the plaintext is larger than n, it should be divided into blocks.

1: $C = P^e \bmod n$, P: Plaintext C: Ciphertext
2: Return C

2.3 Decryption

Bob can use the following steps to decrypt the cipher text message.

1: $P = C^d \bmod n$
2: Return P

The security of RSA is based on the idea that the modulus is so large that is infeasible to factorize in the reasonable time. If Eve (a pseudo name) can factor n and obtain p and q, then she can calculate Φ(n) = (p*1)(q*1), Eve then calculate d, because e and n are public keys.

3. Integer Factoring Algorithms

Algorithms for integer factorization can be split into two groups:

- Special purpose: A special-purpose factoring algorithm's running time depends on the properties of the number to be factored or on one of its unknown factors: size, special form, etc. Exactly what the running time depends on varies between algorithms. e.g. Trial division, Fermat factorization, Pollard ρ-1, VFactor, MVFactor , NF [Pollard, 1974; Ambedkar and Bedi, 2011; Sharma et al., 2012; and Somsuk and Kasemvils, 2013].
- General purpose: The general algorithms are those not targeted at a special class of numbers. e.g. Dixon, Quadratic sieve [Dixon, 1981; and Pomerance, 1985].

Comparison of some special and general purpose factorization algorithms are presented in the Table I.

Table I

Comparison of some special and general purpose factorization algorithms

Algorithm	Type	Technique used	Remarks
Trial division	Special purpose	Division	Useless for large composite number
Fermat factorization	Special purpose	Square root	Suitable for small numbers
Pollard ρ-1	Special purpose	Probabilistic, Fermat Little Theorem	Can't factor all numbers. e.g. n = 65
Pollard ρ	Special purpose	Periodic sequence, Probabilistic	Can't factor all numbers. e.g. n = 21
VFactor	Special purpose	Square root	Can factor all numbers
MVFactor	Special purpose	Square root, Least significant digit	More efficient than Vfactor
NF	Special purpose	Square root	Can't factor all numbers n=p(p+2)
Dixon	General purpose	Legendre Congruence, Random selection	Selecting an optimal bound B-smooth value and random selection of x
Quadratic sieve	General purpose	Legendre Congruence, Polynomial and Sieve interval	Speeds up the process by which x values are found

4. HILL-Fact Method

Hill climbing technique is used to find the factors of an integer number. Hill Climbing is a local search technique. It starts with an initial solution and steadily and gradually generates neighboring successor solutions. If the neighboring state is better than the current state then the neighboring state is considered as the current state. There are different variants of hill climbing namely simple hill climbing, steepest hill climbing, stochastic hill climbing and random restart hill climbing. Fermat tries to rewrite a composite number as a difference of square of two numbers i.e. $N=x^2-y^2$. The heuristic function used to evaluate each node is derived from $N=x^2-y^2$ and the heuristic function is $H(x, y)=N-(x^2-y^2)$. In this paper, (1, 1) is considered as initial node and the production rule to generate successor node is $(x, y) \rightarrow (x+1, y)|(x, y+1)$. HILL-Fact method as follows:

1: Initialize node, (1, 1) and evaluate it

2: If it is a goal node i.e. H(1,1)=0 then stop and return

3: Until a goal node is found (Goal state is reached when H(x, y)=0)

a: Generate the successor node of (x, y) using production rule and evaluate them

b: Compare parent node (x, y) and successor of it and choose the better node

HILL-Fact algorithm is illustrated using a small number N=77. The initial node is (1, 1) and H(1, 1)=77. Since it is not a goal node and H(1, 1)>0, generate successor of it as shown in the Figure 1.

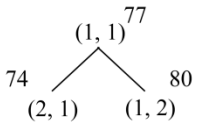

Figure 1: Step 1

The heuristic value of node (2, 1) is 74, which is better than the heuristic value 80 of the node (1, 2), we select (2, 1) as the next node to be expanded as shown in the Figure 2.

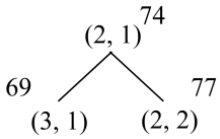

Figure 2: Step 2

Continue this process of generating successors and identifying the best amongst them. A small portion of the tree is shown in the Figure 3.

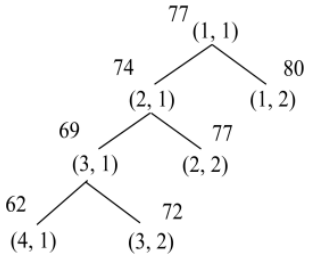

Figure 3: Step 3

4.1 Analysis

This method is suitable for all numbers having two factors. Figure 4 demonstrates the convergence of heuristic function values of the nodes generated for N=77. It is observed that whole process is directed towards the value zero, finally resulting in the solution.

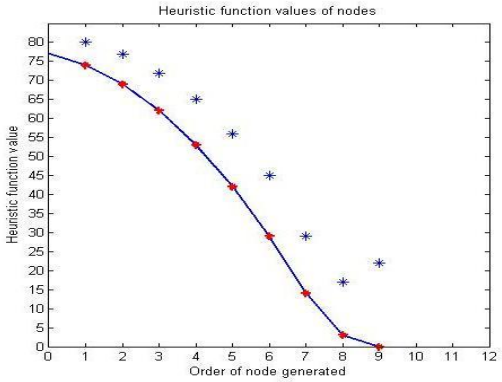

Figure 4: Heuristic function values of nodes

In this method there are two main drawbacks:

- Initial point: In HILL-Fact initial point for all number is (1, 1). But the running time also depends on the initial point. In HILL-Fact there is no initial point selection mechanism.
- No backtracking: The process of finding factors may stuck at any point as shown below:

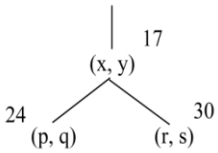

Figure 5: Backtracking problem

In Figure 5, the heuristic function value 24 and 30 are greater than 17. In HILL-Fact there is no mechanism to handle such situation.

5. Conclusion

In this paper a hill climbing technique HILL-Fact to find factors of a number is presented. This method is very simple. From Fermat Factorization Method it is found that $N=x^2-y^2$. From this analysis it can infer that finding factors is searching a point in a two dimensional plan that satisfy the equation: $N=x^2-y^2$. For large number the search space becomes very large. Some conventional method fails to find out the factors for large number. In this paper, AI technique, called Hill climbing is used to find out factors. The further work involves enlarging the scope of the work to take care of mentioned issues.

References

Ambedkar, B. R., & Bedi, S. S. , A New Factorization Method to Factorize RSA Public Key Encryption. *IJCSI International Journal of Computer Science Issues*, **8**(6). 242-247, 2011.

Dixon, J. D. Asymptotically fast factorization of integers. *Mathematics of computation*, **36**(153), 255-260, 1981.

Pollard, J. M., Theorems on factorization and primality testing. In *Mathematical Proceedings of the Cambridge Philosophical Society* (Cambridge University Press), **76,** (03), 521-528,1974.

Pollard, J. M., Monte Carlo methods for index computation ($mod\ p$). *Mathematics of computation*, **32**(143), 918-924,1978.

Pomerance, C., 1985, The quadratic sieve factoring algorithm. In Advances in cryptology, pp. 169-182 (Springer Berlin Heidelberg).

Rivest, R. L., Shamir, A., & Adleman, L. , A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, **21**(2), 120-126, 1978.

Somsuk, K., & Kasemvilas, S., MV Factor: A method to decrease processing time for factorization algorithm. In *Computer Science and Engineering Conference (ICSEC), International* IEEE, 339-342, 2013.

Sharma, P., Gupta, A. K., & Vijay, A. , Notice of Violation of IEEE Publication Principles Modified Integer Factorization Algorithm Using V-Factor Method. In *Advanced Computing & Communication Technologies (ACCT), Second International Conference* IEEE, 423-425, 2012 (January).