EXPLORING A NON FACTOR METHOD OF DECRYPTING THE RSA CODE

J. Nyaga and C. Mwathi
Department of Pure and Applied Mathematics, Jomo Kenyatta University of Agriculture and Technology, Nairobi.
E-mail: cecilia_mwathi@yahoo.com

ABSTRACT

Breaking of the RSA cryptosystem remains an unsolved intriguing mathematical problem. The security of the RSA code rests on the fact that factoring large integers is a hard problem. These are numbers having exactly two large prime factors. Several such numbers with 129 digits or more, known as RSA numbers, have been factored. In spite of this achievement, no progress in breaking the code seems to be forthcoming from the factoring approach. This difficulty arises from availability of a prime number greater than n, where n is a natural number.

In this work, we explore a method that is independent of factoring methods.

With the RSA code, a public key (e, n) is given to the public. We set $p^e \equiv c(\mod n)$ where p is a plaintext word and c is its corresponding ciphertext word. Some secret key $(d, \phi(n))$ (where $\phi(n)$ is the Euler phi function of n) is held by the receiver and is unknown to anybody else. It is known that $ed = 1 \mod \phi(n)$

So by letting the non-reduced value of c be $c + xn$, $x \in \mathbb{Z}^+$

$$p^e = c + xn$$

$$p = (c + xn)^{1/e} (\mod n)$$

We develop mathematical algorithm for calculating the first integral e^th root of $c + xn$. This integer is the required value of p. Using the algorithm we successfully deciphered messages of plaintexts sent in blocks of up to five.

This method requires that the block size is determinable by the decoder. There is need however to develop a system of inferring the length of the blocks used in the plaintext before applying the encryption algorithm in which case the method can be extended to decrypting messages sent in any block length. In the paper we also have two results regarding the choice of the public key in this code.

1. If $p = q$, then n and $\phi(n)$ are both perfect squares and computing the secret key d becomes trivial.

2. If both p and q are twin primes, that is $p = q \pm 2$, n is of the form $q^2 \pm 2q$.

The integer $k = n + 1$ is therefore a perfect square and can be used to estimate the factors of n, hence reducing to the first case.

Key words: Cryptography, cryptanalysis, cryptology, encryption, plaintext, ciphertext, decryption, key, public key cryptography

1.0 INTRODUCTION

RSA is a public- key cryptosystem which uses an algorithm developed in 1977. The acronym stands for three Massachusetts Institute of Technology professors-Ronald L.Rivest, Adi Shamir, and Leonard M. Adleman-the inventors of the system.

1.1 How RSA Works

The RSA is a type of public key cryptosystem. In implementation, one comes up with the following variables:

1. Two distinct large primes *p* and *q* selected at random. The primes *p* and *q* are of roughly the same length *k*, that is *k*-bit long.

2. Compute *n* by the equation $n = pq$ and the Euler Phi function by the equation

$$\phi(n) = (p-1)(q-1)$$

3. Select an integer *e* such that

$$\gcd(e, \phi(n)) = 1$$

4. Compute *d* as the multiplicative inverse of *e*, modulo $\phi(n)$ That is,

$$de = 1 \bmod \phi(n)$$

5. Publish the pair $P = (e, n)$ as RSA public key and keep secret the pair

$S = (d, \phi(n))$ as RSA secret key.

6.For a message *M*(plain text),the encryption function is

$$E(M) = M^e (\bmod n)$$

7.For a cipher text C, the decryption function is

$$D(c) = C^d (\bmod n)$$

As of now, there has not been a proof published that states the infallibility of this method and therefore there is still the possibility of finding the key.

Example:

Pick $p = 11$ and $q = 23$

$n = p * q = 253$

$\phi(n) = (p-1)(q-1) = 220$

Pick some *e* that has no common divisor with phi; $e = 29$ (random number).

$d = 129$ since $29 * 129 = 1 \bmod 220$; this is not at all obvious but nevertheless calculatable.

Encrypt the letter *h* (ASCII value 101); $M = 101$

$C = M^e \bmod n = 50$

Send $C = 50$

Decryption:

$M = C^d \bmod 253 = 101$

$ASCII \;\; 101 = h$

1.2 Security of the Code

If one has the secret key *d*, it is always possible to get the original message from the ciphertex. If an attacker obtained *d,* then it would be possible to read all encrypted messages. One way to do this would be to factor *n* into its prime factors *p* and q and consequently compute $f(n)$ since the public exponent *e* is given, one would obtain the secret exponent *d.*

2.0 PROGRESS MADE

Most researchers have directed their efforts towards resolving the factoring difficulty. As a result the inventors of cryptosystem came up with a factoring challenge which involves factoring certain large numbers called RSA numbers. RSA numbers are composite numbers having exactly two prime factors. In principle, RSA are smaller than the largest known primes but their factorisation is of significance. Cash prizes are awarded to people who successfully factor any RSA number. The RSA numbers were originally picked to have between 100 and 500 digits at interval of 10 decimal digits. They were named according to the number of digits, thus, RSA-100 was a hundred digit long. The 129-digit number known as RSA-129 was factored by Morristown N. J in April 1994 using a method known as quadratic Sieve.

Five years after, H. te Riele completed factorisation of RSA-140 into two 70- digits primes and RSA-155 into two 78-digit primes.

Leo De Velez allegedly broke the RSA cryptosystem. His approach was based on the fact that $f(n) = (p-1)(q-1)$ is always composite. He proceeded to show how one could come out with an integer value *k*, which was not necessarily the secret exponent *d* but nevertheless could work. Rivest (inventor) pointed out a flaw in the Velex method.

Several RSA numbers have been factored since then. The most recent factoring was that of RSA-200 by J. Franke in May 2005. RSA-640 to RSA-2048 remain open with cash prizes ranging between \$20 000 to \$200 000.

Most code busters it seems, have concentrated on factoring *n* as the solve way of breaking the RSA. No progress seems to be forthcoming from this kind of attack.

3.0 RESULTS

3.1 Investigating Key Distance, Band Length and Deciphering Without Using d

3.1 Varying Key Distance

3.1.1 Suppose p = q

This would mean $f(n) = (p-1)(q-1) = (p-1)^2$

$f(n)$ is a perfect square. This makes factoring of $f(n)$ possible.

$\Rightarrow n = pq$ would be a perfect square. This makes factoring of *n* possible since it would be the case of getting the square root of *n*.

That is $p = q = \sqrt{n}$

One would as a result get $p-1$ and $q-1$ hence calculate $f(n)$ and eventually come out with *d*. It would therefore be easy to break the code.

Suppose p and q are twin primes.

If p and q are twin primes they are such that $p = q \pm 2$

Then

$n = pq = q(q \pm 2) = q^2 \pm 2q$

Deduction

If *p* and *q* are twin primes then $(n+1)$ is always a perfect square.

Proof

Let *p* and *q* be twin primes. That is $p = q \pm 2.$

The number between the two primes is $q \pm 1$

So $n = q(q \pm 2) = q^2 \pm 2q$

$k^2 = (q \pm 1)^2 = q^2 \pm 2q + 1$

So the difference between *n* and k^2 is always 1.

Therefore, if p and q are twin primes, then $n+1$ is always a perfect square. To get the two primes, one can first calculate y, where

$$y = \sqrt{(n+1)}$$

Then $p = y - 1$,

$q = y + 1.$

So using twin primes as the secret key puts the code at risk.
One can obtain p and q by taking the square root of n and testing all numbers that are sufficiently close to $\sqrt{n}$ for primallity and then taking the two which differ by two.

Worked Example 1
Suppose $n = 1763$

$$n + 1 = 1764$$

$$y = \sqrt{1764} = 42$$

Therefore, $p = 42 \ +1 = 43$

$$q = 42 \ -1 = 41$$

$$\Rightarrow n = 1763 = 41 \ \times 43$$

Worked Example 2 (Involving large twin primes)

Suppose n = 1 000 000 019 300 000 093 122 499

n+1 = 1 000 000 019 300 000 093 122 500

$$y = \sqrt{1\,000\,000\,019\,300\,000\,093\,122\,500}$$

=1 000 000 009 650

$\therefore p$ =1 000 000 009 650 – 1 = 1 000 000 009 649

q = 1 000 000 009 650 + 1 =1 000 000 009 651

n = 1 000 000 019 300 000 093 122 499

=1 000 000 009 649 * 1 000 000 009 651

3.2 Varying Codeword Length

Deduction
Whenever the block length is greater than the number of digits in n, there is always

a decoding error for all $P_i^s > n$. P_i the ith component of the plaintext.

Results

Let a be the number of digits in *n*, b be the block length, P_i be the ith component of the Plaintext. If $b \geq a$ and $P_i \geq n$ the corresponding decoded component $P_i^|$ is always such that

$P_i^| \neq P_i$ and it satisfies the condition $P_i^| \equiv P_i \bmod n$

Example
Variables:

$p = 131$

$q = 113$

$n = pq = 131 \times 113 = 14803$

$f(n) = (p-1)(q-1) = 130 \times 112 = 14560$

Let $e = 3$

Let $d = 9707$

Secret key $S(f(n), d) = (14560,9707)$

Public key $P(n,e) = (14803,3)$

Suppose $P_i = 51620.$

Then

$$C_i = P_i^e \bmod n = (51620)^3 = 137547911528000 \equiv 12661 \bmod 14803$$

Computing the plaintext using the secret key d we get :

$$P_i = C_i^{\ d} \bmod n = (12661)^{9707} \equiv 12386889108 \ \bmod 14803 \equiv 7211 \bmod 14803$$

This differs from the origin P_i since $P_i > n$. However we notice that

$$51620 \equiv 7211 \bmod 14803$$

3.3 Decryption Independent of Factoring

From our definition of variables, we have

Public key = *(n,e)*

Secret key = $(\mathrm{f}\ (n), d)$

Enciphering: $C = E(P) = P^e \bmod n$

Deciphering: $P = D(C) = C^d \bmod n$

Given *n, e*, and *C*, one can proceed as follows:
For a particular ith component of the cipher text, one determines the relationship

$C_i = P_i^e \bmod n$
…………………………………………………………………………..(1)

which indeed is the encryption function.

C_i is actually the remainder when P_i^e is divided by *n* (from the definition of modulo
arithmetic).
We therefore can rewrite equation 1 as
$P_i^e = xn + C_i$ where $x \in \$!^{+}$ ………………………………………………..(2)
In Public key cryptography one would have to be given the numerical equivalents of the alphabet. Without loss of generality, let us assume our set of numerical conversion is the set of integers (0,1,2,3,4,…………26) to correspond to a space, A,B,C,………Z respectively.

Taken as such P_i is one of the integers 0,1,2,…………26. One can try by brute force all values between 00 and 26 for P_i which the equation 2 above is satisfied for some

integer x. Where the band length is 3, it would be sufficient to try all three digit values for P_i. This by extension can apply for all band lengths, since the band length hints on the number of options to try for P_i; all values of the combinations.
This is an iteration process and therefore can be carried out by a computer.
We come out with such a computer program in Visual Basic.
In our program the algorithm is as follows:

In the relation in equation 2, our program varies the x rather than the $P_i^{'s}$. It takes the values from zero, multiplies it by n, adds C_i to this product and then calculates the e^{th} root of this sum. If the resulting value is an integral value, the program prints this as the P_i. Otherwise it tests the next value of x.
Decoding therefore rests in the solution of the equation

$$P_i = \sqrt[e]{(xn \; + C_i)}$$

Since the encryption function is actually the reverse of the equation we combine both in the program. Hence in the program below, one can get the value of C_i or P_i given either.

3.3.1 Program to Encode/Decode

```
Option Explicit
Dim n As Long ‘product of (p-1)*(q-1) where q are large primes
Dim p As Long ‘variable for plaintext code
Dim  c As Long ‘variable for cipher text code
Dim e As Long ‘prime number relatively prime to n
Dim I As Long ‘counter variable

‘This function will be used to convert plaintext code to cipher text code function
fcnEncode (p,e,n) As Long
    If n = 0 Then n = 1
  fcnEncode = $P^\wedge e \bmod n$
End function
‘this function will be used to convert cipher text code to plaintext code Function
fcnDecode (c,e,n) As Long
  If n = 0 Then n = 1
For I = 1 To 999
If $i^\wedge e \bmod n$ Mod n = c Then GoTo plaintext
   Next i
Exit Function

Plaintext:
   P = i
fcnDecode = p
```

End Function

REFERENCES

Adams W . W. and Goldstein L. J. (1976). Introduction To Number Theory. Prentice-Hall, Englewood Cliffs, New Jersey.

Adleman L., Rivest R. L. and Shamir . A. A method for obtaining Digital signatures and public key cryptosystems. Association for Computing Machinery.

Boneh D and Venkatesan R. (2003): Breaking RSA May Be Easier Than Factorising. Stanford University and Microsoft Research.

Eynden C. V. (1987). Elementary Number Theory. Birkhauser Mathematics Series, Mcgraw-Hill, Inc.

Goldie C. M. and Pinch G. E. (1991): Communication Theory. Cambridge University Press, Cambridge.

Hill R.(1986). A First Course in Coding Theory. Mathematics & Computer Science Series, Clarendon Press, Oxford.

Hungerford T. W. (1990). Abstract Algebra, An Introduction. Saunders College publishing.

Menezes A. J. Oorschot, P.C., &.Vanstone, S. A. (1996): Handbook of Applied Cryptography.Cambridge University Press.

Nyaga J(2004). Exploration Of Non-Factor Methods Of Deccrypting RSA Code. MsC. Dissertation Kenyatta University

Schneier B. (1996). Applied Cryptography. John Willey And Sons Inc.

Stinson D. R. (1995). Cryptography: Theory and Practice. Cambridge University press.

Welsh D. (1988).Codes and Cryptography. Oxford Science Publications, Clarendon Press, Oxford.