The Security of Prime Numbers

Oliver Phillips

11th March 2019

Abstract

In this paper, I shall be exploring prime numbers, their unique properties and how these properties can be applied to keep information secure by the means of encryption algorithms. I shall be focusing on the RSA encryption algorithm due to it being widely used and also being an asymmetric encryption algorithm, as this allows two individuals which have never communicated before to communicate by encrypted means.

1 What are Prime numbers?

The prime numbers were studied at great lengths by many ancient civilisations; the oldest surviving records that are focused entirely upon prime numbers are that of the Ancient Greeks, who studied these numbers in extensive detail [9]. With such research into prime numbers they developed the definition of a prime:

> "A natural number (e.g.$1, 2, 3, 4, \ldots, etc$) is said to be a prime number if it is greater than 1 and cannot be written as a product of two natural numbers that are both smaller than it. Any number that fails to meet this criteria is said to be non-prime [4]".

Using this definition, we can then list some of the smaller prime numbers: $2, 3, 5, 7, 11, 13, 17, 19, \ldots$

Now we have the basic understanding of what a prime number is, the next question that requires great consideration is if there are an infinite amount of prime numbers. One would assume that if there is an endless amount of numbers there will be an infinite amount of prime numbers, but how do we know that this is true? Thankfully, this question has already been proved many centuries ago by a Greek mathematician, Euclid of Alexandria (300 B.C) [1]. Below is a similar proof to that given by Euclid over 2000 years ago, but uses modern concepts and notation:

> Consider any finite list of prime numbers $p_1 = 2 < p_2 = 3 < \cdots < p_n$. Let $P = p_1 \times p_2 \times \cdots \times p_n + 1$ and let p be a prime number dividing P; then p cannot be any of the prime numbers in the list otherwise p would divide the difference $P - p_1 \times p_2 \times \cdots \times p_n = 1$, which is impossible. So, following this, p is a prime number which is not included in the original list [6].

We now have this proof of infinite prime numbers, but as a proof, it feels lacking and inadequate; to help with the understanding of such a proof, I will give an example:

> We consider a list of prime numbers: $\{2, 3, 5, 7, 13, 17, 23\}$.
> Let $P = 2 \times 3 \times 5 \times 7 \times 13 \times 17 \times 23 + 1 = 1,067,430 + 1 = 1,067,431$.
> We now take p which is a prime number which can divide P. If p is a prime number in the list it would divide 1, which is impossible.
> So, p is a prime number which is not in the list and by prime factor decomposition we get the following product:
>
> $P = 1,067,431 = 823 \times 1297$ (it can be checked that both 823 and 1297 are both prime numbers).
>
> From this, we now have a new list: $\{2, 3, 5, 7, 13, 17, 23, 823, 1297\}$ and if we follow the steps in the proof indefinitely, the list of prime numbers will grow indefinitely and therefore there is an infinite number of primes.

2 Is there any use for prime numbers?

In the previous section, we explored what a prime number is and the existence of infinite prime numbers. However, now is the time to ask:

> Are prime numbers useful for anything other than being something interesting to study?

Before we can answer that question, we need to understand one useful property about natural numbers ($\mathbb{N} > 1$). This is that they can be written as a unique product of primes, also known as 'The Fundamental Theorem of Arithmetic'. This means in the form:

$$N = p_1^{n_1} p_2^{n_2} p_3^{n_3} \dots p_k^{n_k} = \prod_{i=1}^{k} p_i^{n_i} \tag{1}$$

So, we take the following numbers for our example: $\{11, 14, 200, 3003\}$.

$$11 = 2^0 \times 3^0 \times 5^0 \times 7^0 \times 11^1 = 11^1 \text{ - (a result which we would expect as 11 is prime).} \tag{2}$$

$$14 = 2^1 \times 3^0 \times 5^0 \times 7^1 \times 11^0 = 2^1 \times 7^1 \text{ - (this is easy to check by hand).} \tag{3}$$

$$200 = 2^3 \times 3^0 \times 5^2 \times 7^0 \times 11^0 = 2^3 \times 5^2 \tag{4}$$

$$3003 = 2^0 \times 3^1 \times 5^0 \times 7^1 \times 11^1 \times 13^1 = 3^1 \times 7^1 \times 11^1 \times 13^1 \tag{5}$$

This property for $\mathbb{N} > 1$ has been proven by Carl Fredrick Gauss in his first book published in 1801, at the age of 24 [5]. There are a variety of proofs regarding the Fundamental Theorem of Arithmetic, which are readily available online, so I shall not include a proof here.

Now that we have introduced the Fundamental Theorem of Arithmetic, we can now answer our most pressing question:

> Are prime numbers useful for anything other than being something interesting to study?

The simple answer is yes. The more complex answer is that prime numbers are widely used in cryptography and encryption algorithms; prime numbers may also be used in other fields. For the rest of this article, we shall be focusing on the use of prime numbers within the RSA encryption algorithm as this is one of the most predominant areas where prime numbers are used.

3 RSA Encryption

RSA (which is named after its creators Ron Rivest, Adi Shamir, and Leonard Adleman) is a form of cryptosystem developed in 1978 [7]. In the five years previous to this, a mathematician who worked for GCHQ (Government Communications Headquarters; British intelligence) developed a similar system but it was deemed dangerous to national security and therefore kept highly classified till 1997 at which point it was declassified [2].

What is a cryptosystem? It is a method of encrypting data for secure transfer over a public domain or network. It works in a similar way to that of a simple cipher that would encrypt a message from a prying classmate for example. In the same way, an encoded message is difficult to understand; RSA does the same thing but against prying computers.

It would be expected that RSA will be much more complex than a simple cipher, as modern day consumer computers are reaching a few 100 billion floating point operations per seconds (FLOPS) while super computers and distributed computing networks are producing even more FLOPS [8][10].

3.1 The RSA Algorithm

The RSA algorithm can be split into four steps: key generation, key distribution, encryption, and decryption.

3.1.1 Step one:

We need to produce our RSA keys; this is done by the following process:

- Pick two different prime numbers p and q.
- Take $n = p \times q$. - (As we have shown there are infinite primes so infinite possibilities for n).
- Compute $\lambda(n) = lcm(\lambda(p), \lambda(q)) = lcm((p-1), (q-1))$.
- Pick a natural number e which satisfies the following conditions: $1 < e < \lambda(n)$ and $gcd(e, \lambda(n)) = 1$.
- Find value d such that $d \equiv e^{-1}$ *modulo* $\lambda(n)$. This is done by solving $d \times e \equiv 1$ *modulo* $\lambda(n)$.

Now we have all the information needed to form our public and private keys:

- Public key = $\{n, e\}$ - This is the information that can be broadcast publicly.
- Private key = $\{d, p, q, \lambda(n)\}$ - These are kept private as these are used in the computation of d.

3.1.2 Step two:

Now that the key generation has been completed, we would distribute the public key through any means (does not have to private/safe) to any person you wish to send an encrypted message to you. This will allow them to encrypt any message they want to keep private from everyone but you.

3.1.3 Step three:

We now need to encrypt the message we would like to transmit but also keep secret, we have to do the following:

- Take our message m and convert it into an integer such that $2 \leq m < n$ (I shall explain more about this during the below example).
- Compute m^e *modulo* $n \equiv c$. c is now our encrypted message and this can now be transmitted.

3.1.4 Step four:

Now that we have received the message c, we would like to decrypt it. To do this we need to do the following:

- c^d *modulo* $n \equiv (m^e)^d$ *modulo* $n \equiv m$ *modulo* n

4 Example of RSA encryption

So, before we can use this encryption algorithm, we need a message we would like to encode and transmit, for this example we shall take the following phrase:

"UoL Mathematical Journal"

It contains 22 characters or 25 characters including spaces. We can use any letter to number cipher but for the sake of simplicity we shall use the most basic:

$$\{a = 2, b = 3, \ldots, y = 26, z = 27\} \tag{6}$$

Other more complex ciphers could be similar to the following, but not limited to:

$$\{A = 2, \ldots, Y = 26, Z = 27, a = 28, b = 29, \ldots, z = 53\} \text{ - Inclusion of capital letters.} \tag{7}$$

$$\{A = 2, \ldots, z = 53, ! = 54, " = 55, \ldots, \% = k\} \text{ - Inclusion of special characters.} \tag{8}$$

4.1 Step zero

We are needing to transfer our 'message' into a plain number format using (6). This will mean our message becomes:

$$\{22, 16, 13; 14, 2, 21, 9, 6, 14, 2, 21, 10, 4, 2, 13; 11, 16, 22, 19, 15, 2, 13\} \tag{9}$$

Given our numerical output, it is now time to decide how we are going to encrypt this data, currently, there are 3 methods which can be used:

1. Encrypting each individual character separately. This is best when giving an example but linguistics can be used when trying to decrypt the message.
2. Encrypting the whole phrase at once. This would be the best method, but this can be quite slow. If the message m is large, then the product of the two primes n would have to be larger. By doing this, it would make the encryption and decryption processes very slow and resource intensive.
3. A combination of the above two methods; encrypting each word or encrypting four consecutive letters at once for example. For this method, there will be a very large variety of ways this can be done. This method would be the best compromise between the two methods above. By doing it in this manner, it would increase the speed of encryption while also making sure that the message is also more secure.

For the simplicity of the example, we shall use the first method as the other two methods require the use of larger numbers and this will make the example much harder to follow.

4.2 Step one

We shall now follow the RSA algorithm described above:

- Pick two prime numbers: $p = 13$ and $q = 19$.
- The product of the two primes: $n = p \times q = 13 \times 19 = 247$.
- Compute the lowest common multiple: $\lambda(n) = lcm(12, 18) = 36$.
- Pick value e such that $gcd(e, \lambda(n)) = gcd(e, 36) = 1 \implies e = 7$
- Find value d such that $d \times 7 = 1 \ modulo \ 36$
  So we want to find the solutions to the following system where both d and g are integers:

$$7 \times d = (36 \times g) + 1 \tag{10}$$

$$d = \frac{(36 \times g) + 1}{7} \tag{11}$$

  The equation (11) can be computed with increasing integers of g till d becomes an integer also, these solutions which are infinite, occur when:

$$g = \{6, 13, 20, \dots\} \tag{12}$$

  For this example, we shall take $d = 31$.

- Public key $= \{n, e\} = \{247, 7\}$
- Private key $= \{d, p, q, \lambda(n)\} = \{31, 13, 19, 36\}$

4.3 Step three

As we have no need to talk about the distribution of the public key, I shall move onto the encrypting process. Just like above, we shall follow the steps outlined in step three above:

- We have already done this in step zero; our message m is the following:

$$\{22, 16, 13; 14, 2, 21, 9, 6, 14, 2, 21, 10, 4, 2, 13; 11, 16, 22, 19, 15, 2, 13\} \tag{9}$$

- To encrypt our message, we now need to do the following for all 22 elements in m. This is done by:

$$c_i \equiv m_i^e \ modulo \ n; \quad i \in 1, 2, \dots, 22 \tag{13}$$

$$c_i \equiv m_i^7 \ modulo \ 247 \tag{14}$$

  When we follow this process, it will look something like this:

$$c_1 \equiv 22^7 \ modulo \ 247 \equiv 2494357888 \ modulo \ 247 \equiv 230 \tag{15}$$

$$c_2 \equiv 16^7 \ modulo \ 247 \equiv 268435456 \ modulo \ 247 \equiv 55 \tag{16}$$

$$\vdots$$

$$c_{22} \equiv 13^7 \ modulo \ 247 \equiv 62748517 \ modulo \ 247 \equiv 143 \tag{17}$$

  Once we have done all 22 computations, our message c will be in the form:

$$\{230, 55, 143, 79, 128, 109, 61, 85, 79, 128, 109, 205, 82, 128, 143, 106, 55, 230, 228, 89, 128, 143\} \tag{18}$$

  This is the message which we can safely transmit, without anyone knowing what our original message is.

4.4 Step four

It's now time to decrypt the message we have received; by following the decryption steps:

- Taking the encrypted message,(18), we can do the following:

$$c_i^d \equiv m_i \ modulo \ n; \quad i \in \{1, 2, \ldots, 22\} \tag{19}$$

$$c_i^{31} \equiv m_i \ modulo \ 247 \tag{20}$$

Unfortunately, it will not be possible to display any number to the power 31 here, as it will be excessively large and therefore cannot be expressed adequately here. With these numbers being so large and being beyond the normal 64-bit representation of most computer calculators, it will take special tools/software to be able to compute these exponent operations accurately.

How I checked that this example is indeed correct, was with the use of the software MATLAB. This involved using an extended toolbox for increasing the numerical precision which I have cited here [3]. This is my original code to conduct the decryption process:

```
function m = decrypt(c,n,d)
% c is our message vector which we want to decrypt; it contains n elements elements
% n is our prime number product (n=pq)
% d is our decryption key
% vpi(k) is a function which allows accurate representation beyond the usual 64 bit precision
    for i = 1:length(c)
        prod = vpi(c(i))^d;      % This produces the accurate representation of  c_{i}^{d}
        m(i) = mod(prod,n);      % This computes the modulo of the product outputting our ...
            value for m(i)
    end
end
```

Using this code, we do indeed get the same message as to that which we encrypted:

$$\{22, 16, 13; 14, 2, 21, 9, 6, 14, 2, 21, 10, 4, 2, 13; 11, 16, 22, 19, 15, 2, 13\} \tag{9}$$

"UoL Mathematical Journal"

5 Limitations of RSA encryption

As with any algorithm, RSA has its limitations. Some of these limitations are:

- The message m has to take a value between 1 and n. An example of why this is the case; if $m = n + 4$. We can say that $m \ modulo \ n \equiv n + 4 \ modulo \ n \equiv 4 \ modulo \ n$. It can be seen that our message $m = n + 4$ becomes 4 under modulo arithmetic and therefore these two messages are very different even before we start the encryption process.
- p and q have to take very large values to increase the time it takes to compute the prime factor decomposition of $n = p \times q$. This is especially important as the RSA algorithm depends on the difficulty of the prime factorisation of n.
- Given large enough numbers of $\{p, q, d, e, c\}$, the time it takes to encrypt and decrypt messages becomes too long and therefore is not suitable for constant data transfer or when the data is needed almost instantly.

Even with such limitations of the algorithm, RSA is widely used for a lot of major data transfer on the internet. A particular use for the RSA algorithm is for the transmission of symmetric cryptosystem keys which allows for faster-encrypted communication between two or more parties.

References

[1] Leonard C. Bruno. *Math and mathematicians: the history of math discoveries around the world*. 2003, p. 125. ISBN: 978-0787638139.

[2] Clifford Cocks. *Note on "Non-secret Encryption"*. 2019. URL: `https://www.gchq.gov.uk/note-non-secret-encryption`.

[3] John D'Errico. *Variable Precision Integer Arithmetic*. 2015. URL: `https://uk.mathworks.com/matlabcentral/fileexchange/22725-variable-precision-integer-arithmetic`.

[4] Anthony Gardiner. *The Mathematical Olympiad Handbook: An Introduction to Problem Solving Based on the First 32 British Mathematical Olympiads 1965-1996*. 1997, pp. 24–26. ISBN: 978-0198501053.

[5] Carl Fredrick Gauss. *Disquisitiones Arithmeticae*. 1801, 1965. ISBN: 0-300-09473-6.

[6] Paulo Ribenbiom. *The New Book of Prime Number Records*. 1995, p. 3. ISBN: 0-387-94457-5.

[7] A. Shamir R.L. Rivest and L. Adleman. *A Method for Obtaining Digital Signatures and Public-Key Cryptosystems*. 1978. URL: https://people.csail.mit.edu/rivest/Rsapaper.pdf.

[8] Seti@home. *CPU performance*. 2019. URL: https://setiathome.berkeley.edu/cpu_list.php.

[9] John Stillwell. *Mathematics and Its History*. 2010, p. 40. ISBN: 978-1441960528.

[10] TOP500. *TOP500 list November 2018*. 2019. URL: https://www.top500.org/list/2018/11/.