IB Mathematics

Extended Essay

Rivest-Shamir-Adleman Cryptography

- How is Rivest-Shamir-Adleman cryptography used today to secure information in the world wide web? -

---

Word Count: 3698

February 7, 2021

Contents

1 Introduction

We have been using cryptography to secure our information throughout the world wide web, where everything and everyone is accessible. When using a social media, such as Facebook, Instagram or Snapchat one becomes curious how your information is secured or stored. With cyber attacks that can leak information, is your information safe when its database is leaked? The significance of cryptography was highlighted in the modern era during World War II, when Alan Turing arguably has single-handedly won the war for his own country by decrypting the Enigma cipher. However, since then cryptography has evolved significantly. Today, we mainly use asymmetric encryption, which means that during encryption two different keys are created. These are called public and private keys. The public key is the encrypting key - it encrypts a message. The private key, on the other hand, decrypts an encrypted message. Moreover, the main encryption method used is end-to-end encryption (EE2E) which allows the sender and the receiver to decrypt and encrypt a message. This is done by the sender generating the encryption key within their own device at that time, whilst the third party software, for example Whatsapp, has no information on the private key. This theoretically ensures that companies cannot track your messages, and even if they get hacked with data leaks, it requires decryption. With the existence and the discovery of such methods of securing data, I became curious on how this process is done. Like other millions of people, I am a frequent user of the world wide web and the internet, and so it is only natural for me to be curious what I am using in my everyday life. As a result, I will be looking into one of the first and most used modern encryptions in the world - Rivest-Shamir-Adlemen (RSA). More specifically, how is the Rivest-Shamir-Adlemenan cryptography used today to secure information in the world wide web?

2 The Purpose of RSA

2.1 Alice and Bob Analogy

1. Suppose that two individuals, Alice and Bob, want to contact each other privately through a long distance. Alice wants to receive some kind of letter from Bob.

2. In order to do this securely, Alice sends a box which can be locked with a regular lock. However, when sending, Alice does not send the keys to the lock.

3. Bob then places this letter inside the box and locks it away with the given lock from Alice.

4. Alice receives this box and unlocks it using her key. She then is able to receive this letter safely. That is, during the process of transferring the box, it is nearly impossible to crack it open.

5. This ”lock” is what we call the **public key**. The key that she uses to unlock the box is called the **private key**. This concept is utilised by the RSA within the process of encryption and decryption of information in the internet. However, in order to make this method possible within computers, they had to develop numerical methods, which resulted in the development of **trapdoor functions**. By definition, these are functions which are easy to compute, but the reverse computation is much more difficult. Therefore, in order to encrypt something, information has to go through the trapdoor function.

This process can be visualised in a simple diagram:

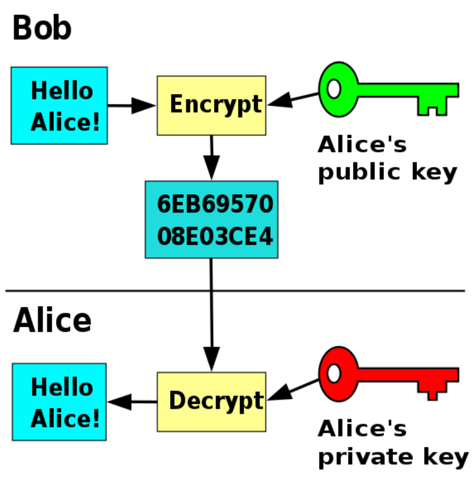


Figure 1: Public Key Encryption Visualisation.

(Göthberg, 2006)

3 Pre-requisites

In order to begin our exploration, we must first understand some of the required mathematics which is used in the RSA method. Unfortunately, the Discrete Mathematics option was removed from the IB syllabus so I will be going through the fundamentals to ensure understanding.

3.1 Definitions

Definition 3.1. Suppose the relation $b = ka$ for $k, a, b \in \mathbb{Z}$. Then we can write this as $a|b$, which means that b is divisible by a or b is a multiple of a (Fannon, Kadelburg, Woolley, & Ward, 2013, p. 13).

Definition 3.2. The least common multiple c of 2 numbers a, b is denoted as $lcm(a, b) = c$. By definition, this is the smallest number c which satisfies $a \mid c$ and $b \mid c$ (Fannon et al., 2013, p. 21).

Definition 3.3. The greatest common divisor c of 2 numbers a, b is denoted as $gcd(a, b) = c$. By definition, this is the largest number c which satisfies $c \mid a$ and $c \mid b$ (Fannon et al., 2013, p. 20).

Definition 3.4. Two numbers a, b are **coprime** or **relatively prime** if and only if $gcd(a, b) = 1$. (Fannon et al., 2013, p. 20).

3.2 Modular Arithmetic

Modular arithmetic of a number can be seen as the 'remainder' cycles of a number when it is divided. For example:

$$\frac{63}{4} = 15\frac{3}{4} \tag{1}$$

There's a remainder 3. Then, the modular arithmetic would be

$$63 \equiv 3 \bmod 4 \tag{2}$$

However, we can apply "cycles" to it. That is, for each congruent relation, we can add and subtract $4k$ (where k is any number such that $k \in \mathbb{Z}$) to 3 or 63 and it will still be considered congruent. For example, 67 will still have remainder 3 and can be expressed as $3 + 4 \cdot 15$, in particular this implies $k = 15$. So then it holds true that

$$63 \equiv 67 \bmod 4 \tag{3}$$

Indeed, this is a result when we add for a positive k. However, we can also subtract by letting k be a negative integer, and as a result we can also obtain negative congruence relations such as

$$-3 \equiv 63 \bmod 4 \tag{4}$$

And since both -3 and 63 have remainders of 3, we can actually interchange these numbers and congruence will still hold true.

$$63 \equiv -3 \bmod 4 \tag{5}$$

So then, we can create a generalisation for some $a, b, c \in \mathbb{Z}$:

$$a \equiv b \bmod c \tag{6}$$

That is, since a and b differ by some multiple c

$$a = kc + b \tag{7}$$

Where $k \in \mathbb{Z}$. As long as these numbers satisfy this equation, then we can say that a is congruent to b modulo c.

3.3 Euclidean Algorithm

The Euclidean Algorithm (Patrick Keef, n.d.) is an efficient method of finding the greatest common divisor of two positive integers, let us call them a and b where $b \neq a$ and $|b| > |a|$. This method uses the division algorithm, and exploits it until the remainder is zero. The theorem states that if $b = ax + r$, then $gcd(a, b) = gcd(b, r)$. The Euclidean algorithm works by initially dividing b by a and writing it in the form with remainder r, that is $b = ax + r$ where $x \in \mathbb{Z}$ obtaining the form shown above. Then, the process is repeated with new divisor r and the new dividend is a, obtaining a new remainder. We repeat this process until there is no remainder. Particularly, this process can be shown as the following:

$$b = ax_1 + r_1 \tag{8}$$

for some x_i where $x_i \in \mathbb{Z}$, and remainder r_i where $r_i \in \mathbb{Z}^+$. Applying the Euclidean Algorithm to compute $gcd(a, b)$ we do

$$\begin{cases} b = ax_1 + r_1 \\ a = r_1x_2 + r_2 \\ r_1 = r_2x_3 + r_3 \\ \vdots \\ r_{i-2} = r_{i-1}x_i + r_i \\ r_{i-1} = r_ix_{i+1} + 0 \end{cases} \tag{9}$$

Where $0 < r_i < r_{i-1}$. Then, by the Euclidean Algorithm, $gcd(a, b) = r_i$. This allows us to create a strict recursive formula which can be used within computer science to find the greatest common divisor of two numbers. This can also be very efficient for numbers a and b whose prime factorisation can be very difficult if they are large. Let us consider an arbitrary example, that is, $gcd(934, 783)$:

$$\begin{cases} 934 = 783 \cdot 1 + 151 & (10) \\ 783 = 151 \cdot 5 + 28 & (11) \\ 151 = 28 \cdot 5 + 11 & (12) \\ 28 = 11 \cdot 2 + 6 & (13) \\ 11 = 6 \cdot 1 + 5 & (14) \\ 6 = 5 \cdot 1 + 1 & (15) \\ 5 = 1 \cdot 5 + 0 & (16) \end{cases}$$

$$\therefore gcd(934, 783) = 1 \tag{17}$$

And let us see why this particularly works with the proof of this algorithm.

Proof. (Patrick Keef, n.d.). Suppose that where $a > r$

$$a \equiv r \bmod b \tag{18}$$

Then, by the definition of modular arithmetic, we could say that

$$a = bk + r \tag{19}$$

$$r = a - bk \tag{20}$$

where $k \in \mathbb{Z}$. Suppose $\exists d$ where $d \in \mathbb{Z}^+$ such that

$$d \mid a \tag{21}$$

$$d \mid b \tag{22}$$

Which also means that it must divide into

$$d \mid a - bk \tag{23}$$

This implies that it could also divide into the left hand side of equation 20, that is

$$d \mid r \tag{24}$$

Which means that d divides into both b and r. By a similar argument, if d divides into r and b instead, then by equation 19 it must also divide into the left hand side, that is

$$d \mid a \tag{25}$$

Therefore, if d divides into r and b, then it also divides into b and a. In particular, this d could the greatest common divisor, thus $gcd(a, b) = gcd(r, b)$. □

3.4 Bézout's Identity and Extended Euclidean Algorithm

Bézout's identity (Brilliant.org, n.d.-a) states that $\forall a, b, d$ where $a, b, d \in \mathbb{Z}$ and $gcd(a, b) = d$, then $\exists x, y$ where $x, y \in \mathbb{Z}$ such that

$$ax + by = d \tag{26}$$

And a solution to Bézout's identity could be interpreted as a 'reverse' process of the Euclidean Algorithm in a sense. This 'reverse' process is also called the Extended Euclidean Algorithm. It would be best to demonstrate the reverse process as an example. Let us consider our two arbitrary integers from earlier, i.e. $a = 934$ and $b = 783$. Then, by Bézout's identity $\exists x, y$ where $x, y \in \mathbb{Z}$

$$934x + 783y = 1 \tag{27}$$

And to find a pair of solution we will consider substituting values back from our computation of the greatest common divisor, beginning with equation 15

$$6 - 1 \cdot 5 = 1 \tag{28}$$

Substituting in equation 14 for 5

$$6 - 1 \cdot (11 - 1 \cdot 6) = 1 \tag{29}$$

$$\tag{30}$$

Expanding and collecting terms

$$2 \cdot 6 - 1 \cdot 11 = 1 \tag{31}$$

Substituting in equation 13 for 6

$$2 \cdot (28 - 2 \cdot 11) - 1 \cdot 11 = 1 \tag{32}$$

$$2 \cdot 28 - 5 \cdot 11 = 1 \tag{33}$$

And we repeat this process until we can express 28 and 11 in terms of 934 and 783

$$1 = 2 \cdot 28 - 5 \cdot (151 - 5 \cdot 28) \tag{34}$$

$$1 = 27 \cdot 28 - 5 \cdot 151 \tag{35}$$

$$1 = 27 \cdot (783 - 5 \cdot 151) - 5 \cdot 151 \tag{36}$$

$$1 = 27 \cdot 783 - 140 \cdot 151 \tag{37}$$

$$1 = 27 \cdot 783 - 140 \cdot (934 - 783) \tag{38}$$

$$1 = 167 \cdot 783 - 140 \cdot 934 \tag{39}$$

hence for our example a possible solution is $y = 167$ and $x = -140$ by the Extended Euclidean Algorithm.

3.5 The Fundamental Theorem of Arithmetic

Beforing proceeding with the Fundamental Theorem of Arithmetic, we first need to consider the following lemma:

Lemma 3.1. *Euclid's Lemma. For any two integers b and c, suppose $a \mid bc$. Then if a is coprime to c, then a divides b.*

Proof. (*Euclid's Lemma*, n.d.). Let

$$a \mid bc \tag{40}$$

Let us assume $gcd(a, c) = 1$. That is, $a \nmid c$. By Bézout's Identity we can write this as

$$1 = xa + yc \tag{41}$$

for some x, y where $x, y \in \mathbb{Z}$. Multiplying everything by b

$$b = bxa + byc \tag{42}$$

Notice that $a \mid bxa$ and $a \mid byc$ from equation 40. Since a divides into right hand side, then it must also divide into left hand side, that is

$$a \mid bxa + byc \implies a \mid b \tag{43}$$

□

The Fundamental Theorem of Arithmetic states that for every integer n which is $n > 1$ can be factored into primes and it is unique.

Proof. (Brilliant.org, n.d.-d). We will begin the proof by first showing that $\forall n$ where $n \in \mathbb{Z}$, $n > 1$ can be factored into primes. By induction, consider base case $n = 2$. 2 is already a prime, therefore it has been factored. Assume true for

$$2, 3, 4, 5, \dots, k \tag{44}$$

Consider $n = k + 1$. If $k + 1$ is prime, then it is already factorised. However, if $k + 1$ is not a prime then $k + 1 = p \cdot N$ for some $N < k$ where $N \in \mathbb{Z}$ and p where p is the smallest prime factor. By our assumption, since $1 < N < k$, N can be factored into more primes. This implies that $k + 1 = p \cdot N$ can be factored into more primes. Therefore, as it holds true for our base case, and when assumed true for some $n = k$, it is also follows true for $n = k + 1$ where $k \in \mathbb{Z}$ and $k > 1$. By the principle of mathematical induction, it follows true $\forall n \in \mathbb{Z}$ where $n > 1$. □

The second proof will show the uniqueness of this factorisation.

Proof. (Brilliant.org, n.d.-d). Assume that an integer n can be factorised in two different ways, with primes p and q listed in ascending order that is, $p_1 \leq p_2 \leq \dots \leq p_{i-1} \leq p_i$ and $q_1 \leq q_2 \leq \dots \leq q_{i-1} \leq q_i$, therefore the prime factorisation of n can be written as

$$n = p_1 \cdot p_2 \cdot \ldots \cdot p_{i-1} \cdot p_i \tag{45}$$

$$n = q_1 \cdot q_2 \cdot \ldots \cdot q_{i-1} \cdot q_i \tag{46}$$

Then, since all primes are coprime with each other, and by Lemma 3.1 we can say that p_1 divides into some number in n where $n = q_1 \cdot q_2 \cdot \ldots \cdot q_{i-1} \cdot q_i$ and since all q are prime, the only way it will divide is if $q_1 = p_1$. Using the same argument for all prime numbers, we can say that $\forall i$ that $q_i = p_i$ □

3.6 Fermat's Little Theorem

Fermat's Little Theorem states that if p is a prime and a is any integer then (Fannon et al., 2013, p. 63)

$$a^p \equiv a \bmod p \tag{47}$$

Which is equivalent to saying

$$p|(a^p - a) \tag{48}$$

And therefore this implies that if p is a prime and a is not a multiple of p, we could divide everything by p to obtain

$$a^{p-1} \equiv 1 \bmod p \tag{49}$$

Proof. (Brilliant.org, n.d.-c) Consider the congruence $a^p \equiv a \pmod{p}$
With base case $a = 1$, we obtain that

$$1 \equiv 1 \bmod p \tag{50}$$

Which is true for our base case.
Let us assume true for some $a = k$ where $k \in \mathbb{Z}^+$

$$k^p \equiv k \bmod p \tag{51}$$

Let us write this assumption as the following instead for convenience

$$k^p + 1 \equiv k + 1 \bmod p \tag{52}$$

Therefore we are looking to show that it holds true for $a = k + 1$ given the above assumption, more specifically, we are looking to show that the following equation holds true:

$$(k+1)^p \equiv k + 1 \bmod p \tag{53}$$

By binomially expanding left hand side we get

$$(k+1)^p = \sum_{i=0}^{p} \binom{p}{i} k^{p-i} 1^i \tag{54}$$

$$= k^p + pk^{p-1} + \frac{p(p-1)}{2!} k^{p-2} + \cdots + \frac{p(p-1)}{2!} k^2 + pk + 1 \tag{55}$$

And by applying $\mathrm{mod} p$ to equation 55 we notice that all the middle terms cancel as $p \mid p!$, leaving us with $k^p + 1$. However, notice that this is only possible because $p \nmid i!$ and $p \nmid (p-i)!$. Otherwise, if there existed a p in the denominator, the p would cancel and thus the final expression would not be divisible by p. Hence, we obtain

$$(k+1)^p \equiv k^p + 1 \bmod p \tag{56}$$

And finally substituting our assumption (equation 52) to equation 56 we show that equation 53 holds true as well. Therefore we get

$$(k+1)^p \equiv k + 1 \bmod p \tag{57}$$

As it holds true for base step, and when assumed true for some $a = k$ where $k \in \mathbb{Z}^+$, it holds true for $a = k + 1$. Then, by the principle of mathematical induction, the statement holds true for all positive integers. However, by the definition of modulo, what holds for positive integers will also hold for all integers. Therefore, it holds true for all integers. □

4 Understanding RSA

4.1 Euler's Totient Function

Euler's function is denoted by

$$\varphi(n) \tag{58}$$

for some $n \in \mathbb{Z}^+$, and it counts the amount of coprimes the number n has. That is, it counts the amount of N which satisfy $gcd(N, n) = 1$ (Brilliant.org, n.d.-b) where $\forall N, N \in \mathbb{Z},\ 1 \leq N < n$. For example, consider when $n = 16$:

$$\varphi(16) \tag{59}$$

$$gcd(1,16) = 1 \qquad gcd(5,16) = 1 \qquad gcd(9,16) = 1 \qquad gcd(13,16) = 1 \tag{60}$$

$$gcd(2,16) = 2 \qquad gcd(6,16) = 2 \qquad gcd(10,16) = 2 \qquad gcd(14,16) = 2 \tag{61}$$

$$gcd(3,16) = 1 \qquad gcd(7,16) = 1 \qquad gcd(11,16) = 1 \qquad gcd(15,16) = 1 \tag{62}$$

$$gcd(4,16) = 4 \qquad gcd(8,16) = 8 \qquad gcd(12,16) = 4 \qquad gcd(16,16) = 16 \tag{63}$$

Counting the amount of coprimes we get that

$$\varphi(16) = 8 \tag{64}$$

From this we could also deduce two identities:

$$gcd(1, n) \equiv 1 \tag{65}$$

This implies that all numbers will have at least 1 coprime. And

$$gcd(n, n) \equiv n \tag{66}$$

Which implies that no number can ever be established such that $\varphi(n) = n$ except $n = 1$. However, if our number is some prime number p, then it follows that

$$\varphi(p) = p - 1 \tag{67}$$

Proof. By Theorem 3.5 all numbers $x \geq 2$ where $x \in \mathbb{Z}$ can be factored into primes. If x is prime, then its only factor is x. Hence, for all q where $2 \leq q < p$ for $q \in \mathbb{Z}$ can be factored into primes. Moreover, p is some prime number. This implies that all q can be factored into n arbitrary amount of primes, $a_1, a_2, \ldots, a_n$. Suppose, for a contradiction, that $\exists q$ which has at least p as one of its prime factors, meaning $a_1 = p$. This implies

$$q = p \cdot \ldots \cdot a_n \tag{68}$$

where $a_n \geq 2$

However, as q is a product of p and possibly other primes, then $q \geq p$ only. This implies $q \not< p$. This is a contradiction.

$$\therefore \text{There is no such } q \text{ that shares a factor with } p$$

This implies that for all q

$$gcd(q, p) = 1 \tag{69}$$

And $\exists p - 2$ of q numbers from the defined inequality. Recalling that $\varphi(n)$ counts the amount of coprimes between $1 \leq N < n$ means that we are missing the following:

$$gcd(1, p) = 1 \tag{70}$$

From our identity in equation 65 and adding this to our count we finally get that

$$\varphi(p) = p - 1 \tag{71}$$

$\square$

With this unique property, it becomes evident that $\varphi(n)$ can achieve the highest output if n are prime. I have plotted the Totient function in Wolfram Mathematica so we can see this:

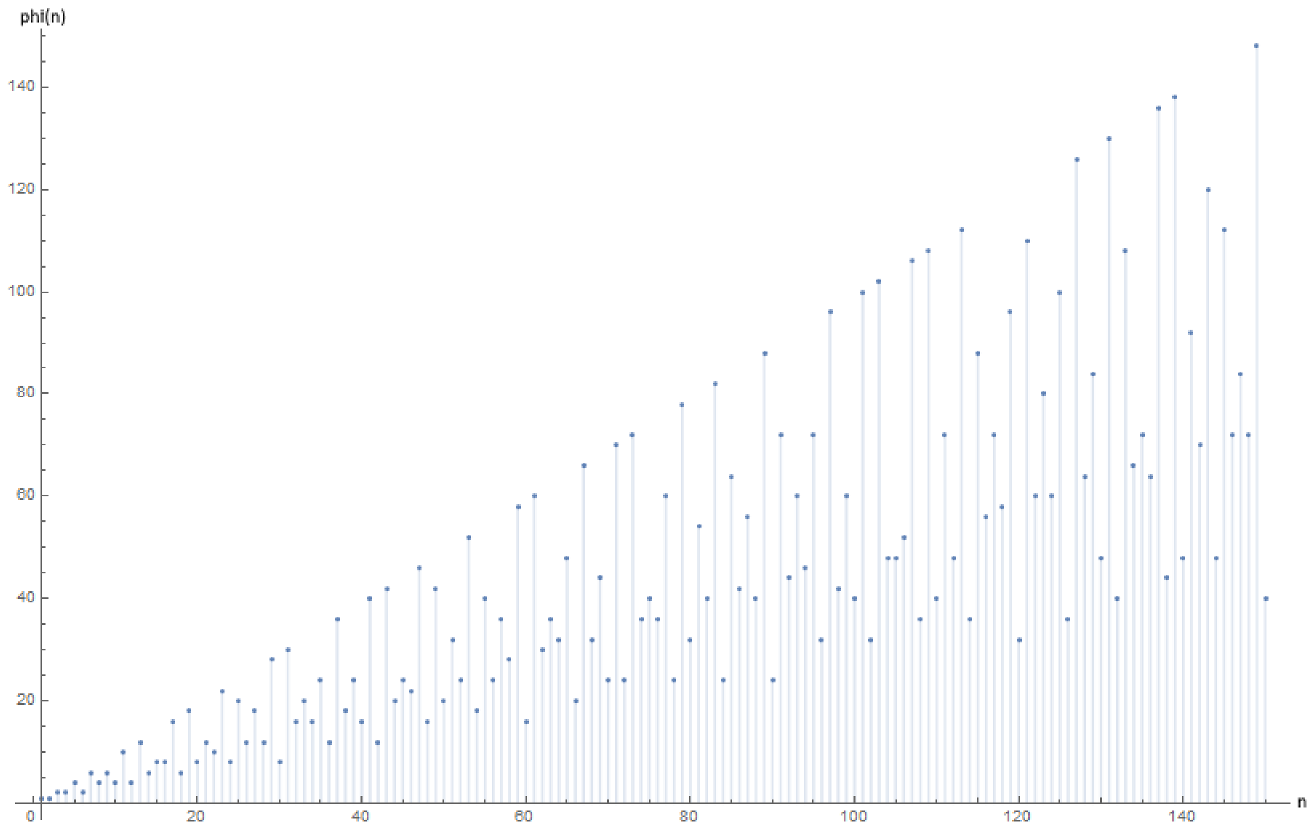


Figure 2: Graph of $\varphi(n)$ for $0 \leq n \leq 150$.
(self-made figure) (Inc., 2016)

Moreover, there exists an identity that states if p and q are coprime, then

$$\varphi(pq) \equiv \varphi(p)\varphi(q) \tag{72}$$

Unfortunately the proof of this multiplicative behaviour is out of the scope of this essay due to ring theory.

And this is the Euler's Totient function, one of the key components of RSA encryption.

4.2 Euler's Theorem

Euler's Theorem states that if two numbers a and n are coprime, then

$$a^{\varphi(n)} \equiv 1 \bmod n \tag{73}$$

Fermat's Little Theorem is actually a special case of the Euler's Theorem when $n = p$ for some prime number p.

Proof. Recall that Fermat's Little Theorem states that for prime p and $p \nmid a$ we can have

$$a^{p-1} \equiv 1 \bmod p \tag{74}$$

Recall from equation 71 that

$$\varphi(p) = p - 1 \tag{75}$$

Therefore if $n = p$ then we get that by substituting $\varphi(p)$ into equation 73

$$a^{p-1} \equiv 1 \bmod p \tag{76}$$

Which is Fermat's Little Theorem. □

Moreover, for the proof of Euler's Theorem, we will first have to define a reduced residue system $\mod n$

Definition 4.1. (Ikenaga, 2019). A reduced residue system $\mod n$ is a set A such that

$$A \in \{a_1, a_2, a_3, \ldots, a_{\varphi(n)}\} \bmod n \tag{77}$$

and if we define a set A where

$$A \in \{a_i, \ldots, a_{\varphi(n)}\} \bmod n \tag{78}$$

and

$$A \in \{a_j, \dots, a_{\varphi(n)}\} \bmod n \tag{79}$$

it must follow that, for it to be a reduced residue system, $a_i \neq a_j \bmod n$ or $i \neq j$. That is, the a are unique in $\bmod n$. Another property that must follow true for a set to be a reduced residue system is that $\forall i$,

$$gcd(a_i, n) = 1 \tag{80}$$

And this creates a set A such that all remainders of an arithmetic are coprime with n, and they are all unique, that is, they do not repeat themselves in cycles. In fact, it could be simply thought of a list of numbers which satisfy Euler's Totient Function. That is, whilst Euler's Totient Function **counts** the amount of coprimes $\forall N$ between $1 \leq N < n$ where $N \in \mathbb{Z}$ for that particular n, the reduced residue system **lists** all N which are coprime.

Let us consider an example of a reduced residue system of $\bmod 16$, a number which we have previously used to compute $\varphi(16)$. Then the set A with $\bmod 16$ could be defined as

$$A \in \{1, 3, 5, 7, 9, 11, 13, 15\} \bmod 16 \tag{81}$$

And with the definition of the residue system being unique as stated, this set could also, for example, be defined as

$$A \in \{17, -13, 21, 23, 41, -21, -3, 31\} \bmod 16 \tag{82}$$

However, the set A cannot be defined as

$$A \in \{-15, 17, -13, 21, 23, 41, -21, -3, 31\} \bmod 16 \tag{83}$$

because -15 and 17 are not unique, that is, they both could be expressed as $1 \equiv \bmod 16$. Moreover, an interesting property is $\forall m$ where $m \in \mathbb{Z}$ such that $gcd(m, n) = 1$, the set A for

$$A \in \{ma_i, \dots, ma_{\varphi(n)}\} \bmod n \tag{84}$$

is still a reduced residue system. The formal proof of this property is beyond the scope of this Extended Essay. Knowing these facts, we can proceed to the proof of Euler's Theorem.

Proof. (Ikenaga, 2019). Let the set A be a reduced residue system such that

$$A \in \{a_1, \dots, a_{\varphi(n)}\} \bmod n \tag{85}$$

And we know that if $gcd(m, n) = 1$ then we also have a reduced residue system such that

$$A \in \{ma_1, \dots, ma_{\varphi(n)}\} \bmod n \tag{86}$$

Since these are still reduced residue sets of $\bmod n$, we can in fact create an equivalence relation where

$$ma_1 \cdot \dots \cdot ma_{\varphi(n)} \equiv a_1 \cdot \dots \cdot a_{\varphi(n)} \bmod n \tag{87}$$

$$\tag{88}$$

Factoring out m, knowing that m exists $\varphi(n)$ times

$$m^{\varphi(n)}(a_1 \cdot \dots \cdot a_{\varphi(n)}) \equiv a_1 \cdot \dots \cdot a_{\varphi(n)} \bmod n \tag{89}$$

$$\tag{90}$$

And dividing everything by $a_1 \cdot \dots \cdot a_{\varphi(n)}$

$$m^{\varphi(n)} \equiv 1 \bmod n \tag{91}$$

Which is Euler's Theorem. □

4.3 RSA Encryption Process

RSA exploits the idea that it is very difficult to factorise numbers into their respective primes. Because there is no clear method, a computer has to do a large amount of computations in order to figure out what the prime factors are of a number. When numbers get very large, this becomes a very tedious and a very long process. This is called **time complexity**, and is in fact a huge study of Computer Science. However, for this extended essay, we will just be using the idea that the larger the number in bits, the significantly harder it becomes to factor it into its primes. The RSA Laboratories has even given a list of numbers which are a product of two primes, and made it a public challenge to factorise them in 1991 (*RSA numbers*, n.d.). For example, the 232 digit number which was factored in 2009 is estimated to take 2000 years to factor in an old processor (Kleinjung et al., 2010, p. 14). However, they were able to cut the time by using various special algorithms designed for those particular numbers. The trapdoor function of RSA is then clear, the utilisation of the product of primes.

1. Therefore, to see this process (Mollin, 2002, p.60-63), we begin with picking 2 distinct random primes, let us denote them p and q. Let us denote the product of these 2 numbers as n.

   $$pq = n \tag{92}$$

   And thus it follows that

   $$\varphi(n) = (p-1)(q-1) \tag{93}$$

2. We select some number e which will denote the encryption key where $e \in \mathbb{N}$, $1 < e < \varphi(n)$ and $gcd(e, \varphi(n)) = 1$. We will also select some number d where $1 < d < \varphi(n)$ and $d \in \mathbb{N}$ which is the decryption key, that must satisfy

   $$ed \equiv 1 \bmod \varphi(n) \tag{94}$$

3. The public encryption key e and n is then published to the public for encryption. In order to encrypt a message with ciphertext M where $M < n$ into encrypted message C, we must use the following relation

   $$C \equiv M^e \bmod n \tag{95}$$

4. And if receiving the message to decrypt the ciphertext C we must use the following equivalence relation

$$C^d \equiv M \bmod n \tag{96}$$

Now let us consider an example using American Standard Code for Information Interchange (ASCII). ASCII Decimals allow us to code characters of computers in numbers. To see which printable characters correspond to numbers see Appendix A. For example, we will encrypt the word "egg". This translates into the following using ASCII:

$$101103103 \tag{97}$$

Now picking 2 prime numbers, let

$$p = 77773, q = 83339 \tag{98}$$

$$\therefore n = 6481524047 \tag{99}$$

$$\therefore \varphi(6481524047) = 77772 \cdot 83338 \tag{100}$$

$$\varphi(6481524047) = 6481362936 \tag{101}$$

Now we pick e which satisfies $gcd(e, 6481362936) = 1$. Let $e = 257$, d then must satisfy

$$257d \equiv 1 \bmod 6481362936 \tag{102}$$

This could be written using the definition of modulo for some $k \in \mathbb{Z}$

$$257d + 6481362936k = 1 \tag{103}$$

This equation could in fact be solved by the Extended Euclidean Algorithm as we have done in our example, however, since the number is too large we will use technology to find the value of d (see Appendix B for the python code). This obtains us $d = 1437500729$. Thus we can now encrypt our message using our e value

$$2005725653 \equiv 101103103^{257} \bmod 6481524047 \tag{104}$$

And now we can decrypt this using our value of d

$$2005725653^{1437500729} \equiv 101103103 \bmod 6481524047 \tag{105}$$

Which indeed translates back into ”egg”. And as such, it is possible to see how things are encrypted and decrypted. Now let us see why this mathematically works.

4.4 Mathematics of RSA

Let us see how and why the RSA works mathematically by utilising Fermat’s Little Theorem as well as Euler’s theorem for two special cases of M and n. First, we will consider when our M is coprime with n.

Proof. (Ireland, 2002). As defined, we know

$$ed \equiv 1 \bmod \varphi(n) \tag{106}$$

Rewriting this using the definition of modular arithmetic

$$ed = k\varphi(n) + 1 \tag{107}$$

We also know that

$$C \equiv M^e \bmod n \tag{108}$$

Raising everything to the power of d we obtain

$$C^d \equiv M^{ed} \bmod n \tag{109}$$

And if we substitute our equation 107 for ed into 109 we get

$$C^d \equiv M^{k\varphi(n)+1} \bmod n \tag{110}$$

$$C^d \equiv M(M^{\varphi(n)k}) \bmod n \tag{111}$$

We know $gcd(M, n) = 1$. Then, by Euler's Theorem, we can obtain $M^{\varphi(n)} \equiv 1 \bmod n$ and $M^{\varphi(n)k} \equiv 1^k \bmod n$. Thus we obtain after applying this to the top equation

$$C^d \equiv M(1^k) \bmod n \tag{112}$$

$$C^d \equiv M \bmod n \tag{113}$$

for our unique M. □

However, what if our M is not coprime with one of our prime numbers, let us say q? Then we could consider the following:

Proof. (Ireland, 2002). Let us assume that our number q divides into M, that is $q \mid M$, therefore by the definition of modular arithmetic we have

$$0 \equiv M \bmod q \tag{114}$$

Raising everything to the power of $1 + k\varphi(n)$ where k is some integer such that $k \in \mathbb{Z}$ we get

$$0^{1+k\varphi(n)} \equiv M^{1+k\varphi(n)} \bmod q \tag{115}$$

$$0 \equiv M^{1+k\varphi(n)} \bmod q \tag{116}$$

And since 0 is congruent to $M \bmod q$ as defined, then

$$M \equiv M^{1+k\varphi(n)} \bmod q \tag{117}$$

One of the main properties of RSA is that $M < n$. Since p and q are distinct primes, we can deduce that $p \nmid q$. Moreover, since the product $pq = n$, we can say that if $q \mid M$, then $p \nmid M$. This also implies that

$gcd(M, p) = 1$, therefore by Fermat's Little Theorem we could say that

$$M^p \equiv M \bmod p \tag{118}$$

Dividing by M we obtain

$$M^{p-1} \equiv 1 \bmod p \tag{119}$$

Let us now raise everything to the power of $q - 1$

$$M^{(p-1)(q-1)} \equiv 1^{q-1} \bmod p \tag{120}$$

$$M^{(p-1)(q-1)} \equiv 1 \bmod p \tag{121}$$

Particularly, we know that the power is in fact the definition of $\varphi(n)$

$$M^{\varphi(n)} \equiv 1 \bmod p \tag{122}$$

And we if we raise this to the power of k where k is some number such that $k \in \mathbb{Z}$

$$M^{k\varphi(n)} \equiv 1^k \bmod p \tag{123}$$

$$M^{k\varphi(n)} \equiv 1 \bmod p \tag{124}$$

Multiplying everything by M we get

$$M^{1+k\varphi(n)} \equiv M \bmod p \tag{125}$$

And thus we get that

$$\begin{cases} M^{1+k\varphi(n)} \equiv M \bmod p \\ M^{1+k\varphi(n)} \equiv M \bmod q \end{cases} \tag{126}$$

From the definition of the modular arithmetic we could derive that $p \mid M^{1+k\varphi(n)} - M$ and

$q \mid M^{1+k\varphi(n)} - M$, we then could indeed say that $pq \mid M^{1+k\varphi(n)} - M$ as the dividend's prime factorisation must then at least consist of the primes p and q. Using this argument we could express this as

$$M^{1+k\varphi(n)} \equiv M \bmod pq \tag{127}$$

And since $pq = n$, then

$$M^{1+k\varphi(n)} \equiv M \bmod n \tag{128}$$

From the definition of ed in RSA, we have

$$ed \equiv 1 \bmod \varphi(n) \tag{129}$$

Which by the definition of the arithmetic modulo we could express it as

$$ed = 1 + k\varphi(n) \tag{130}$$

for $k \in \mathbb{Z}$. And in particularly, if we define our encrypted message C to be

$$C \equiv M^e \bmod n \tag{131}$$

If we raise everything to the power of d, then

$$C^d \equiv M^{ed} \bmod n \tag{132}$$

Then, if we substitute equation 130 into equation 128 we get

$$M^{ed} \equiv M \bmod n \tag{133}$$

And indeed, from congruence relations of equations 132 and 133, we obtain

$$C^d \equiv M \bmod n \tag{134}$$

for our unique M. $\square$

Both of these proofs show the correctness of RSA, that is, our encryption and decryption are unique within our process, exposing RSA's mathematical secrets utilising theorems which were originally used to create this encryption process.

5 Conclusion

Indeed, it seems that the Rivest-Shamir-Adleman method is a very clever manipulation of various theorems in order to derive unique integers C and M from the public and the private key e and d respectively. By utilising number theory in the most creative ways, Rivest, Shamir and Adleman came up with one of the most commonly used mathematical algorithms globally. Whilst RSA is one of the most prevalent cryptographical methods used today within the world wide web, it is important to note that many new cryptographical methods were developed since then. For example, one of the newest developed methods is Elliptic Cryptograhy, which is commonly used in cryptocurrency such as Bitcoin. In fact, Elliptic Cryptography is harder to decrypt, and contradictingly to what one would think, it takes less memory to encrypt and decrypt using this method. It would be interesting to see as to how this cryptographical method works. Nevertheless, RSA is still one of the easiest methods to implement within the universe of Computer Science. It was wonderful to see how each block of mathematics leads to harder and steeper steps, which was especially evident with RSA, given that each theory is derived from another fundamental theory (or as some refer to such 'supporting theories', a lemma). However, whilst computers today do struggle to factorise large numbers to their prime numbers, this may become irrelevant in the future. It is important to note that with the introduction of quantum computers, factorisation of numbers into their primes will become almost instantaneous, rendering this particular cryptographical method useless. As such, it is also important to find more complex cryptographical methods which can still appeal to a larger $O(n)$ for quantum computers - an intriguing field of study. Today, a regular desktop computer would take 6.4 quadrillion years (Thakkar, 2018) to crack a 2048-bit RSA private key! It seems that information on the internet is indeed secure, as the complexity of decrypting RSA is high. Whilst a 2048-bit RSA private key is not commonly used within the world wide web, today's computers are still struggling in decrypting the RSA efficiently. Moreover, the era of quantum computers is afar and as a result the RSA should suffice for a long period of time that is to come before quantum computers are properly introduced for public use.

References

Algorithm Implementation/Mathematics/Extended Euclidean algorithm [Encyclopedia Article]. (n.d.). Retrieved 2020-09-18, from https://en.wikibooks.org/wiki/Algorithm_Implementation/Mathematics/Extended_Euclidean_algorithm

ASCII Table – Printable Characters [Text]. (n.d.). Retrieved 2020-09-17, from http://facweb.cs.depaul.edu/sjost/it212/documents/ascii-pr.htm

Brilliant.org. (n.d.-a). *Bezout's Identity* [Website]. Retrieved 2020-09-19, from https://brilliant.org/wiki/bezouts-identity/

Brilliant.org. (n.d.-b). *Euler's Totient Function* [Website]. Retrieved 2020-09-11, from https://brilliant.org/wiki/eulers-totient-function/

Brilliant.org. (n.d.-c). *Fermat's little Theorem* [Website]. Retrieved 2020-09-12, from https://brilliant.org/wiki/fermats-little-theorem/

Brilliant.org. (n.d.-d). *Fundamental Theorem of Arithmetic* [Website]. Retrieved 2020-09-22, from https://brilliant.org/wiki/fundamental-theorem-of-arithmetic/

Euclid's Lemma. (n.d.). Retrieved 2020-09-22, from https://artofproblemsolving.com/wiki/index.php/Euclid%27s_Lemma

Fannon, P., Kadelburg, V., Woolley, B., & Ward, S. (2013). *Mathematics Higher Level Topic 10 - Option: Discrete Mathematics* [Book]. Cambridge University Press.

Göthberg, D. (2006, August 07). *Public Key Encryption* [Website]. Retrieved 2020-09-14, from https://commons.wikimedia.org/wiki/File:Public_key_encryption.svg

Ikenaga, B. (2019). *Euler's Theorem* [Text]. Retrieved 2020-10-18, from http://sites.millersville.edu/bikenaga/number-theory/eulers-theorem/eulers-theorem.html

Inc., W. R. (2016). *Mathematica, Version 11.0* [Software]. Retrieved from https://www.wolfram.com/mathematica (Champaign, IL, 2016)

Ireland, D. (2002). *RSA Theory* [Text]. Retrieved 2020-09-17, from https://www.di-mgt.com.au/rsa_theory.html

Kleinjung, T., Aoki, K., Franke, J., Lenstra, A., Thomé, E., Bos, J., … Zimmermann, P. (2010). *Factorization of a 768-bit RSA modulus.* Cryptology ePrint Archive, Report 2010/006. Retrieved from https://eprint.iacr.org/2010/006

Mollin, R. A. (2002). *RSA and PUBLIC-KEY CRYPTOGRAPHY* [Book]. Chapman & Hall.

Patrick Keef, D. G. (n.d.). *The Euclidean Algorithm* [Text]. Retrieved 2020-09-24, from https://www.whitman.edu/mathematics/higher_math_online/section03.03.html

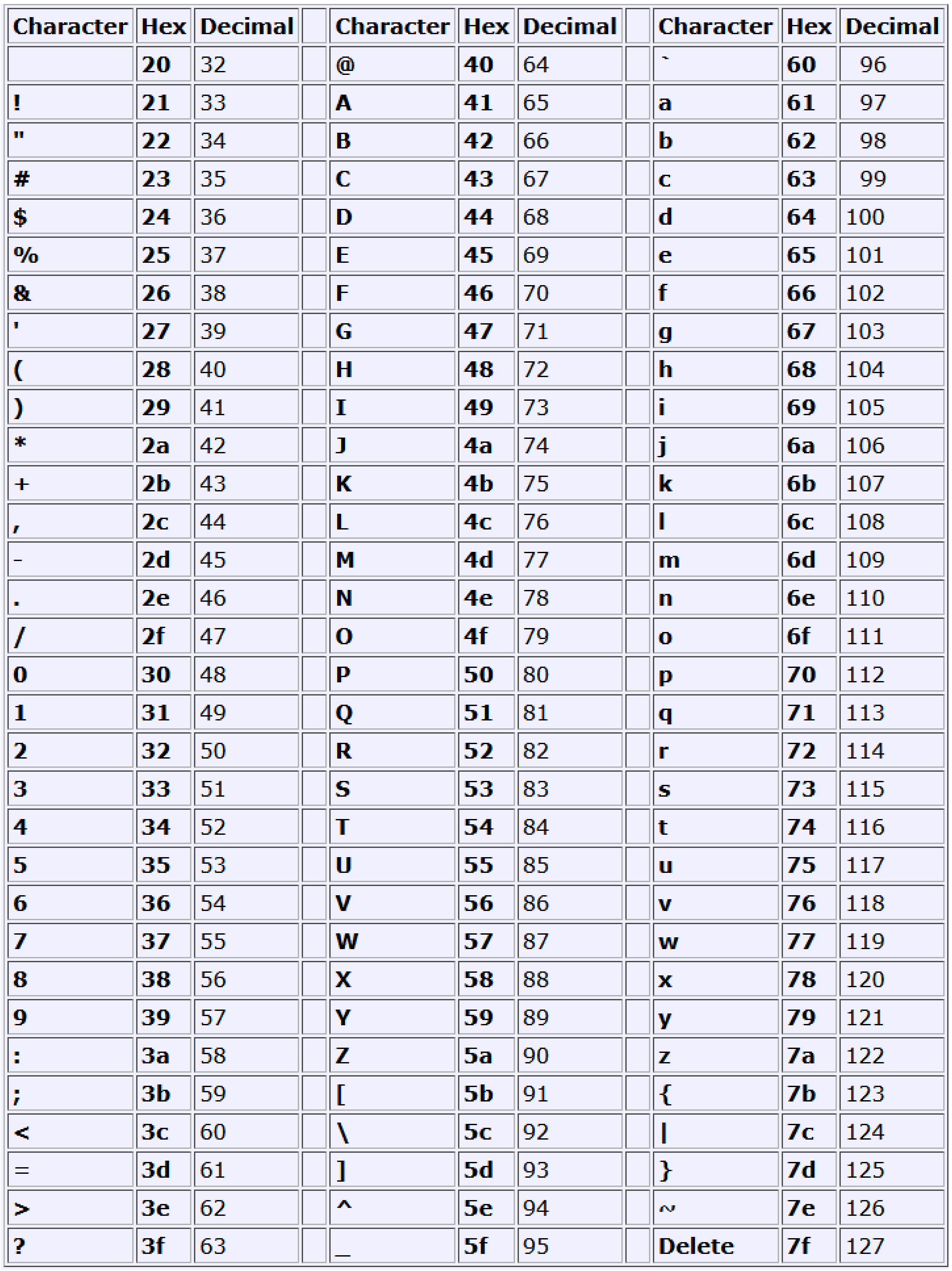
RSA numbers [Encyclopedia Article]. (n.d.). Retrieved 2020-09-15, from https://en.wikipedia.org/w/index.php?title=RSA_numbers&oldid=978484950

Thakkar, J. (2018). *Quantum Computing's Threat to Public Key Cryptography.* Retrieved 2020-10-24, from https://www.thesslstore.com/blog/quantum-computings-threat-public-key-cryptography-need-worry/

Appendices

A ASCII Table

(*ASCII Table – Printable Characters*, n.d.)



Character	Hex	Decimal		Character	Hex	Decimal		Character	Hex	Decimal
	20	32		@	40	64		`	60	96
!	21	33		A	41	65		a	61	97
"	22	34		B	42	66		b	62	98
#	23	35		C	43	67		c	63	99
$	24	36		D	44	68		d	64	100
%	25	37		E	45	69		e	65	101
&	26	38		F	46	70		f	66	102
'	27	39		G	47	71		g	67	103
(	28	40		H	48	72		h	68	104
)	29	41		I	49	73		i	69	105
*	2a	42		J	4a	74		j	6a	106
+	2b	43		K	4b	75		k	6b	107
,	2c	44		L	4c	76		l	6c	108
-	2d	45		M	4d	77		m	6d	109
.	2e	46		N	4e	78		n	6e	110
/	2f	47		O	4f	79		o	6f	111
0	30	48		P	50	80		p	70	112
1	31	49		Q	51	81		q	71	113
2	32	50		R	52	82		r	72	114
3	33	51		S	53	83		s	73	115
4	34	52		T	54	84		t	74	116
5	35	53		U	55	85		u	75	117
6	36	54		V	56	86		v	76	118
7	37	55		W	57	87		w	77	119
8	38	56		X	58	88		x	78	120
9	39	57		Y	59	89		y	79	121
:	3a	58		Z	5a	90		z	7a	122
;	3b	59		[	5b	91		{	7b	123
<	3c	60		\	5c	92		|	7c	124
=	3d	61		]	5d	93		}	7d	125
>	3e	62		^	5e	94		~	7e	126
?	3f	63		_	5f	95		Delete	7f	127

B Python Code

(*Algorithm Implementation/Mathematics/Extended Euclidean algorithm*, n.d.)

```
def egcd(a, b):
    if a == 0:
        return (b, 0, 1)
    else:
        g, y, x = egcd(b % a, a)
        return (g, x - (b // a) * y, y)


def modinv(a, m):
    g, x, y = egcd(a, m)
    if g != 1:
        raise Exception('modular inverse does not exist')
    else:
        return x % m


print(modinv(257,6481362936))
```