Int. J. Math. And Appl., 8(4)(2020), 21–27
ISSN: 2347-1557
Available Online: http://ijmaa.in/

International Journal of *Mathematics And its Applications*

Cryptography and Number Theory

Kalan Nguyen[1,*]

1 Clemson Avenue, Saratoga, California, United States.

Abstract: Number theory is the branch of pure mathematics which studies positive whole numbers also called natural numbers and integers. The goal of number theory is to discover relationships between numbers that may not be apparent at face value. Cryptography is the study of encryption and decryption of information. By encrypting information the data can be transmitted to the intended recipient securely across the internet (insecure network). The use of numbers in cryptography is inextricably linked.

Keywords: Cryptography, Number theory, Encryption and decryption.

1. Introduction

This research paper is dedicated to exploring the principles of number theory and applying them to cryptography. Cryptography is an important aspect of communicating securely with others. It allows for others to maintain their privacy. It prevents interceptors and unlikely holders of your message from being able to read what you are sending to the endpoint. It protects information and a variety of information like passwords, keys, names, bank accounts, etc. This raises the question of what makes cryptography function, how cryptography works, and how number theory is linked to it. In these circumstances, we will be demonstrating the algorithms of cryptography that are formulated with the programming language, Python, because it has an understandable readability and shows the principles, concepts, and applications to the wonderful world of cryptography.

Consider the following scenario:

Viet and Lee are close friends and study together. They both like to communicate by handing each other pieces of paper with writing, however, they want to keep their messages to each other. Many other students may happen to have the paper and could read everything that the two are sending to each other. To solve this problem, Lee and Viet use a pattern and shift the position of the letters so the messages they send do not make sense to the people who intercept them but makes sense to them. Lee sends Viet a message saying "Jq uqi oanf fq cfiju oyfv km?" which translates to "Do you want to study with me?" for Viet and Lee. The message was intercepted and read by multiple students since Lee passed the paper down to the back of the classroom from the front. None of the students knew what the paper meant at all except for Viet and Lee.

What Viet and Lee used was cryptography, a field that studies the techniques of secure communication. The use of cryptography was originally meant for communicating with other people in wars or conflicts. This was used to prevent the

* *E-mail: scoutkalan@gmail.com*

other belligerents from reading messages which could give away large amounts of information. Today, cryptography is used to encrypt messages to prevent interceptors from reading them and is practiced all over the internet. We will be introducing mathematical techniques in number theory which can be further applied to cryptography in encrypting messages.
We refer the reader to [1–3] for more details about number theory and to [4–6] for more details about cryptography.

2. Divisibility and Greatest Common Divisor

Definition 2.1. *The Greatest Common Divisor is the largest integer that can divide each integer in a set of integers.*

For example, there are two numbers, 15 can divide 45 because $45 = 15 \cdot (3)$.
The *Greatest Common Divisor* which is commonly known as the "GCD" is one of the key components in Euclid's algorithm in number theory. The greatest common divisor of two or more integers is the largest positive that can divide the integers. Let a and b be two integers where there is a unique integer q and a positive integer r namely that $r < |b|$ and $a = qb + r$. In this case, q is the quotient of a divided by b and r is the remainder of the division.
A perfect example of this would be if we divided 50 by 12, the quotient is 4 so that must mean that the remainder is 2 because $50 = (4)\,12 + 2$ and $0 \leq 4 < 12$.
Another thing to take into account is that b divides a which is equivalent to the remainder of a divided by b equals 0.

Fact 2.2. *Henceforth, the equation provided is valid for integers a and b.*

$$gcd\,(a,\ b) = gcd\,(b,\ a\,\%\,b) \tag{1}$$

An example of using this equation would be $a = 40$ and $b = 25$. Equation (1) would then deem that $gcd\,(40,\ 25) = gcd\,(25,\ 15)$, but since the result is still not complete, we recur the equation again until $a\%b = 0$. This leads to $gcd\,(25,\ 15) = gcd\,(15,\ 10) \rightarrow gcd\,(15,\ 10) = gcd\,(15,\ 10)$. We then arrive to our result that $gcd\,(40,\ 25) = 5$.

Fact 2.3. *Given two integers a and b, there are two integers such that u and v exist.*

$$gcd\,(a,\ b) = ua + vb \tag{2}$$

In this example, $a = 40$, $b = 25$, $gcd\,(40,\ 25) = 5$ and $5 = (2)\,40 + (-3)\,25$. As a result, $u = 2$ and $v = -3$. Another thing to note is that $5 = (7)\,40 + (-11)\,25$ with $u = 7$ and $v = -11$ is also a valid option. This implies that the pair of numbers which are u and v are not unique.

3. Euclidean Algorithm

The Euclidean algorithm is a powerful computing method that can calculate the greatest common divisor of two integers. In the first chapter, we learned how to make a basic version of the Euclidean algorithm. In this chapter, we will make a more complicated version of the Euclidean algorithm to better understand how it works and apply it to other subjects.
For this case, let a and b be defined as two integers greater or less than zero. The sequence of these integers taking the remainder into account as $r_0, r_1, ..., r_n$ will be defined:

$$r_0 = a, r_1 = b, \text{for } i \geq 1, \text{ if } r_i \neq 0, \text{ then } r_{i+1} = r_{i-1}\,\%\,r_i \tag{3}$$

The variable, n is the integer defined attributable to the fact that $r_{n-1} \neq 0$ and $r_n = 0$. Equation (3) implies that there is an increment in the remainder in the modulo operation until $r_n = 0$ by virtue of $r_2 = r_0 \,\%\, r_1 \;\rightarrow\; r_3 = r_1 \,\%\, r_2$ and so forth. An example could be if a = 20 and b = 15, then $r_0 = 20$ and $r_1 = 15$, so $r_2 = 20 \,\%\, 15 = 5$ and $r_3 = 15 \,\%\, 5 = 0$. From **Fact 2.3**, there is an indication that $r_{n-1} = r_2 = gcd(20,\ 15) = gcd(a,\ b)$.

Fact 3.1. *$r_0, r_1, ..., r_n$ is the sequence defined by Equation (3) where n is the positive integer in which $r_{n-1} \neq 0$ and $r_n = 0$, then $gcd(a,\ b) = r_{n-1}$.*

This sequence helps supply an algorithm to compute the *Greatest Common Divisor* for two numbers. An implementation for this algorithm exists in Python and the way it works is that a function is declared as *gcd* and takes two parameters which are *a* and *b*; returns the *Greatest Common Divisor*.

```
def gcd(a, b):
    if (b == 0):
        return a
    else:
        return gcd(b, a % b)
```

For example, $gcd(20,\ 15)$ returns the number 5. This simple implementation returns the *Greatest Common Divisor* by recursively looping until variable *b* becomes zero since $a \,\%\, 0 = 0$ is final the stopping point.
Let a and b be two integers greater than zero and $r_0, r_1, ..., r_n$ be the sequence which was defined in Equation (3). The sequences $u_0, u_1, ..., u_n$ and $v_0, v_1, ..., v_n$ are defined in the following:

$$u_0 = 1, u_1 = 0, \quad \text{for } 1 \leq i < n, \quad u_{i+1} = u_{i-1} - \left(\left\lfloor \frac{r_{i-1}}{r_i} \right\rfloor\right) \cdot u_i \tag{4}$$

$$v_0 = 0, v_1 = 1, \quad \text{for } 1 \leq i < n, \quad v_{i+1} = v_{i-1} - \left(\left\lfloor \frac{r_{i-1}}{r_i} \right\rfloor\right) \cdot v_i \tag{5}$$

Remember that $u_0a + v_0b = r_0$ because $u_0a + v_0b = 1r_0 + 0r_1 = r_0$ which means that $u_1a + v_1b = r_1$ and $u_1a + v_1b = 0r_0 + 1r_1 = r_1$. The fact that $u_2a + v_2b = r_2$ is because

$$\left(u_0 - \left(\left\lfloor \frac{r_0}{r_1} \right\rfloor\right) \cdot u_1\right) \cdot a + \left(v_0 - \left(\left\lfloor \frac{r_0}{r_1} \right\rfloor\right) \cdot v_1\right) \cdot b = (u_0a + v_0b) - \left(\left\lfloor \frac{r_0}{r_1} \right\rfloor\right)(u_1a + v_1b) = r_0 - \left(\left\lfloor \frac{r_0}{r_1} \right\rfloor\right) \cdot r_1 = r_0 \,\%\, r_1 = r_2.$$

Fact 3.2. *For $0 \leq i < n$, there exists $u_ia + v_ib = r_i$. Because $r_{n-1} = gcd(a,\ b)$, there exists an equation such that:*

$$u_{n-1}a + v_{n-1}b = gcd(a,\ b) \tag{6}$$

One way this could be put together is by sequencing it mathematically so it gives someone an idea on how it will be computed.

$u_0,\ v_0,\ r_0 \ =\ 1,\ 0,\ a$

$u_1,\ v_1,\ r_1 \ =\ 0,\ 1,\ b$

$i = 1$

while $r_i \ \neq \ 0$

$q = \left\lfloor \frac{r_{i-1}}{r_i} \right\rfloor$

$u_{i+1} = u_{i-1} - q \ \cdot \ u_i$

$v_{i+1} = v_{i-1} - q \ \cdot \ v_i$

$r_{i+1} = r_{i-1} \,\%\, r_i$

$i = i + 1$

return r_{i-1}, u_{i-1}, v_{i-1}

Using this, we can define a Python function, **gcdr** which takes two parameters a and b, and returns three linked integers which are g, u, v to ensure that $g = gcd(a, b)$ and $ua + vb = g$.

```
def gcdr(a, b):
    up, vp, rp = 1, 0, a
    uc, vc, rc = 0, 1, b
    while rc != 0:
        q = rp // rc
        up, uc = uc, up - q * uc
        vp, vc = vc, vp - q * vc
        rp, rc = rc, rp - q * rc
    return rp, up, vp
```

Take for example, **gcdr**(24, 16) returns three integers which are 8, 1, -1 where $8 = gcd(24, 16)$ and $(1)24 + (-1)16 = 8$.

4. Primes & Prime Factorization

Definition 4.1. *A positive integer, p, is prime if $p > 1$, and the only positive integers that can divide p are 1 and p.*

For example, 5 is a prime number but 10 is not because 5 divides 10—$5 \neq 10$, $5 \neq 1$.

Fact 4.2. *The integer, n, will be positive where there is a unique determinate sequence of prime numbers, $p_1 < p_2 < ... < p_k$ and another unique determinate sequence of positive integers $n_1, n_2, ..., n_k$ thus $p_1^{n_1}, p_2^{n_2}, ..., p_k^{n_k}$. This is known as the* Prime Factorization *of n.*

Some useful examples of this could be $64 = 4^3$, $75 = 5^2(3), 11 = 11$. These are the factorization of some integers.

5. Congruences

Definition 5.1. *There will be two integers, a and b, a positive integer, m. Integer a is congruent to b* modulus *m which is then denoted to $a \equiv b (mod\, m)$ if m divides $a - b$.*

Take for example, 12 is congruent to 2 *modulus* 5 $\rightarrow$ $12 \equiv 2 (\mathrm{mod}\, 5)$ because 5 divides 10 which was found by $12 - 2$.

Fact 5.2. *There are two integers, a and b, and m, a positive integer.*

$$a \equiv a \,\%\, m (mod\, m) \tag{7}$$

Using Equation (7), an example would be $17 \equiv 17 \,\%\, 4 (\mathrm{mod}\, 4)$ because $17 \,\%\, 4 = 1$ and 4 divides 16 which was found by $17 - 1$.

Fact 5.3. *Let a_1, a_2, b_1, and b_2 be integers; m will be a positive integer. Then assume that $a_1 \equiv a_2 (mod\, m)$ and $b_1 \equiv b_2 (mod\, m)$.*

$$a_1 + b_1 \equiv a_2 + b_2 (mod\, m) \;\; and \;\; a_1 b_1 \equiv a_2 b_2 (mod\, m) \tag{8}$$

For example, $7 \equiv 2 (\mathrm{mod}\, 5)$, $3 \equiv 8 (\mathrm{mod}\, 5)$ and $10 = 7 + 3 \equiv 10 = 2 + 8 (\mathrm{mod}\, 5)$ and $21 = 7(3) \equiv 16 = 2(8) (\mathrm{mod}\, 5)$.

6. Solving $ax \equiv c(mod\, m)$ for x

Considering that a, c are integers and m is a positive integer, the objective is to solve for x so that $ax \equiv c(\text{mod}\, m)$ and $1 \leq x < m$.

Fact 6.1. *Have $d = gcd(a, m)$ assuming that d divides c. Let u and v be integers so that $ua + vm = d$ and $x = \frac{uc}{d}\, \%\, m$. Then have $ax \equiv c(mod\, m)$ and $1 \leq x < m$.*

The reason this fact is valid is because it results from a string of congruences $\rightarrow ax = a(uc/d\%m) \equiv a \cdot \frac{uc}{d} \equiv (a \cdot \frac{uc}{d} + v \cdot \frac{mc}{d}) \equiv \frac{c}{d} \cdot d \equiv c(\text{mod}\, m)$.
Using the previous theorem, we can create a Python function that takes parameters, a, c, and m called **axcm** so that $gcd(a, m)$ divides c; returns x as the single solution to $ax \equiv c(\text{mod}\, m)$ with $1 \leq x < m$.

```
def axcm(a, c, m):
    g, u, v = gcdr(a, m)
    x = ((u * c) // g) % m
    return x
```

For example, **axcm**(2, 3, 5) returns the integer 4 so $2x \equiv 3(\text{mod}\, 5)$ that $x = 4$ because $2(4) - 3 = 5 = 5(1)$ thus 5 divides $2(4) - 3$.

7. Euler's Formula

Euler's Formula well-known as *Euler's Phi Function* or *Euler's Totient Function*. Given the prime factorization that n is a positive integer, there is also a range of $1 \leq k \leq n$ for which the *Greatest Common Divisor*, $gcd(n, k)$.

Fact 7.1. *Let p and q be two new variables which are different primes that $m = pq$. Let a be an integer so that $gcd(a, m) = 1$. Then $a^{(p-1)(q-1)} \equiv 1(mod\, m)$. This specific case which gives a more general result is known as* Euler's Formula.

To demonstrate this fact, let $p = 3$ and $q = 5$. There are powers of a^k *modulus* 15 for $1 \leq k \leq 8$ for all a that $gcd(a, 15) = 1$ and $1 \leq a < 15$. Then $a^{(p-1)(q-1)} = a^8 \equiv 1(\text{mod}\, 15)$ for all numbers of a which abides to Euler's formula.

8. Solving Euler's Formula: $x^k \equiv b(\textbf{mod}\, m)$ for x

Fact 8.1. *Three integers, b, m, and k need satisfy three requirements.*

1. *$m = pq$ with p and q being prime.*
2. *$gcd(b, m) = 1$*
3. *$gcd(k, (p-1)(q-1)) = 1$*

There will be two integers, u and v so there is $uk + v(p-1)(q-1) = 1$. Then the solution of another equation below.

$$x^k \equiv b(\text{mod}\, m) \tag{9}$$

Which satisfies $1 \leq x < m$ for:

$$x = (b^u)\, \%\, m \tag{10}$$

To comprehend the validity of Fact 10, first observe that $x^k = ((b^u) \,\%\, m)^k \equiv ((b^u))^k (\text{mod}\, m) \equiv (b^{uk})(\text{mod}\, m)$ which is:

$$x^k = b^{uk} (\text{mod}\, m) \tag{11}$$

Since $gcd(b,\ m)$ is equal to 1, from Fact 9, there exists $b^{(p-1)(q-1)} \equiv 1(\text{mod}\ m)$ so $1 \equiv 1^v \equiv (b^{(p-1)(q-1)})^v \equiv b^{v(p-1)(q-1)} (\text{mod}\, m)$, e.g.

$$1 \equiv b^{v(p-1)(q-1)} (\text{mod}\, m) \tag{12}$$

The product to the left of the Equations (11) & (12) are congruent to the product of the side to the right of *modulus* m. Which means that we have $x^k \equiv b^{uk} b^{v(p-1)(q-1)} \equiv b^{uk+v(p-1)(q-1)} \equiv b^1 \equiv b (\text{mod}\, m)$ which proves that Fact 10 is valid.
For example, we have $k = 3$, $b = 2$, $p = 3$, and $q = 5$. In this instance, $(p-1)(q-1) = 8$ because $((3)-1)((5)-1) = 8$. And that we have $3(3) + (-1)8 = 1$ for the *Greatest Common Divisor*. Thus, $u = 3$ as per Fact 10. Also take note that $m = 15$ and Fact 10 gives the solution to $x^3 \equiv 2(\text{mod}\ 15)$ respectively with $1 \leq x < m$; $x = (2^3) \,\%\, 15 = 8$. Which means $8^3 \equiv 64(8) \equiv 4(8) \equiv 32 \equiv 2(\text{mod}\ 15)$.
The following Python function, **xtoKEqb** takes four integers as parameters, k, b, p, q, which fulfill the three conditions in Fact 10, and returns the output, x, as an integer which goes hand in hand with $x^k \equiv b(\text{mod}\, m)$ and $1 \leq x < m$.

```
def xtoKEqb(k, b, p, q):
    if gcdr(b, m)[0] != 1 or gcdr(k, (p - 1) * (q - 1))[0] != 1:
        return False
    m = p * q
    d, u, v = gcdr(k, (p - 1) * (q - 1))
    return (b ** u) % m
```

This function first defines m as $p \cdot q$ as specified earlier in Fact 10. To verify that it's valid an *if* statement checks if $gcd(b,\ m) = 1$ or $gcd(k,\ (p-1)(q-1)) = 1$ which are essential conditions as mentioned in Fact 10; if none of the two conditions checked are equal to 1, then the function will return *False*, indicating that it cannot proceed with given the input that does not work.

9. Applications to Cryptography

Reimagine the scenario between Viet and Lee. Lee wants to send a message to Viet who is the receiver of it. The message will be a number because they can map text to numbers and contrarily. Viet will decrypt the message once he receives it. Both use the following scheme:

1. Both select two large prime numbers, p and q.
2. They let $m = pq$ and Lee uses k so that $gcd(k,\ (p-1)(q-1)) = 1$ in his scheme. He makes k and m public that means everyone aside from Viet and Lee knows what the value of k and m are.
3. Lee wants to send the message b, a number to Viet. The limitation to b is that $1 \leq b < m$. So therefore, Lee has to compute $c = (b^k) \,\%\, m$ and instead sends b to Viet.
4. Because Viet knows the values of k, p, and q and $gcd(k,\ (p-1)(q-1)) = 1$, he can solve for x with Euler's function, $x^k \equiv c(\text{mod}\, m)$ with $1 \leq x < m$. However, $b^k \equiv c(\text{mod}\, m)$ and $1 \leq b < m$. Since x's solution is unparalleled, they have $x = b$ so Viet can satisfactorily decode the message from Lee's endpoint.

5. Viet can successfully decrypt the message at his endpoint and absolutely no one except Viet and Lee will know what the message being sent is.

Since the values of p and q are required to solve $x^k \equiv c(\mathrm{mod}\, m)$ for x with $1 \leq x < m$, and Viet is the only person aside from Lee who knows these values, Viet is the only one who knows how to decrypt the incoming message. This method that Viet and Lee are using is commonly known as *RSA*.
A Python function, **encrypt** demonstrates this by encrypting the message.

```
def encrypt(b, k, m):
    return (b ** k) % m
```

Furthermore, another function, **decrypt** will decrypt the message. Parameters, p and q are not passed because both the sender and receiver know the needed primes before the message is sent. This knowledge of the primes, p and q among Viet and Lee serves as their secret key that helps them decrypt the message since it's only known to them. Because p and q are only known to Viet and Lee, we can assume that there is a key holding two values for p and q.

```
def decrypt(k, c):
    p, q = key[0], key[1]
    return xtoKEqb(k, c, p, q)
```

As specified earlier, c is used in the function **xtoKEqb** as it serves as the second argument derived from the function, **encrypt**. As the functions are pieced together, they serve a very beneficial use for Viet and Lee, so they can send each other messages without the worry of an outsider reading their messages.

10. Discussion

In this article, we identify the algorithms powered by number theory which are extensively used in cryptography, and in RSA. From the help of mathematics specifically number theory, we can compute efficient, effective, and organized algorithms in Python to help us better understand number theory but ultimately, cryptography. Cryptography is one of the many benefits of applying mathematics and number theory to the digital world.

References

[1] K. Ireland and M. Rosen, *A classical introduction to modern number theory*, Second Edition, Springer-Verlag, New York, (1990).

[2] H. Stark, *An introduction to number theory*, Cambridge, The MIT Press, (1978).

[3] J. Silverman, *A friendly introduction to number theory*, Fourth Edition, Pearson, New York, (2011).

[4] S. Padhye, R. Sahu and V. Saraswat, *Introduction to cryptography*, Boca Raton, CRC Press, (2018).

[5] S. Rubinstein-Salzedo, *Cryptography*, Springer Nature, Switzerland, (2010).

[6] M. Omar, *Number Theory Toward RSA Cryptography: in 10 Undergraduate Lectures (Discrete Mathematics)*, Volume 1, CreateSpace Independent Publishing, Scotts Valley, (2017).