The Evolution of Cryptography Through Number Theory

Dawson Shores

November 30, 2020

Abstract

Cryptography, the science of disguising messages in order to increase the security of the message, has been in practice for thousands of years. The ability to send messages in secret has been influential throughout history. Another influence in history is cryptanalysis, the technique of uncovering encrypted messages without knowing the decryption key. What started as simply wrapping paper around a stick has evolved into complex internet encryption using mathematics. The ciphers that were used thousands of years ago, while adequate for their time, would necessarily be replaced by more secure ciphers. As more advanced ciphers would be broken, new ones would need to be created. As a result, cryptography is always changing. One key element of this change is the inclusion and progression of mathematics. From simpler arithmetic such as addition and multiplication, to the use of more advanced techniques such as matrix operations, modular arithmetic, and discrete logarithms, a wide variety of mathematics is incorporated into cryptography. A specific field of mathematics that is essential to cryptography is number theory. While there are various ciphers that use number theory, public key ciphers are one of the most important in today's society. Public key ciphers are essential in modern day security for the internet and credit card transactions. This paper describes some of the earlier ciphers that use number theory, and then focuses on different types of public key ciphers such as RSA and ElGamal, as well as the Diffie-Hellman Key Exchange.

1 Introduction to Cryptography

The need for secret communication has been around for centuries. There are two main types of secret communication, steganography and cryptography. Steganography is when the sender of a message would hide the existence of the message [1]. An example of steganography is when people would use invisible ink that could only be read when heated. However, cryptography is more widely known and used. Cryptography is when the message is disguised instead of hidden. In cryptography, there are important terms that need to be defined. The original message, also known as the plaintext, is encrypted and sent away as the ciphertext, which is then decrypted by the recipient. A cipher, or cryptosystem, is what is used to encrypt and decrypt the messages [2]. The sender and recipient will usually have an agreed upon key. An encryption key is used to create the ciphertext while a decryption key is used to decrypt the ciphertext into the original message [3]. There are two types of cipher keys, symmetric/private key and asymmetric/public key. In symmetric key systems the sender and recipient know the key, while in public key systems the encryption key is known but it is computationally infeasible to determine the decryption key if it is not already known [3]. Symmetric key systems are older and there is a wider variety. The first major symmetric key system was the substitution cipher.

2 Shift Ciphers

2.1 Introduction

One of the earliest substitution ciphers was the Caesar shift cipher, used by Julius Caesar [1]. Caesar would replace the original letters of the message with the letters that are three letters down in the alphabet. A description of how the cipher works follows:

Suppose the plaintext “Math” is to be encrypted using the Caesar cipher. Table 1 gives the corresponding ciphertext alphabet.

plaintext	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
ciphertext	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c

Table 1

Note that M is mapped to *p*; A is mapped to *d*; T is mapped to *w* and H is mapped to *k*. Thus, "Math" is encrypted to *pdwk*.

2.2 Encryption

The shift cipher is a special type of monoalphabetic substitution cipher, in which a single cipher alphabet is used throughout the entire encryption process. In shift ciphers, the number that each letter of the plaintext is shifted by is called the key, which we will refer to as *k*. In the Caesar cipher the key, *k*, is 3. In shift ciphers each plaintext letter corresponds to a number as follows:

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

q	r	s	t	u	v	w	x	y	z
16	17	18	19	20	21	22	23	24	25

Table 2

Let m denote the numerical value of the plaintext letter, and c denote the numerical value of the ciphertext letter. The plaintext is converted letter by letter to the ciphertext, *c*, by the following encryption function:

$$c \equiv m + k \pmod{26}, \qquad \text{where } k \in \mathbb{Z}_{26}.$$

For example, the encryption algorithm for a Caesar cipher is $c \equiv m + 3 \pmod{26}$, and we encrypt "math" as follows:

plaintext	M	A	T	H
m	12	0	19	7
$c \equiv m + 3 \pmod{26}$	$12 + 3 \equiv 15$	$0 + 3 \equiv 3$	$19 + 3 \equiv 22$	$7 + 3 \equiv 10$
ciphertext	*p*	*d*	*w*	*k*

Thus the ciphertext *pdwk* is achieved.

2.3 Decryption

In a shift cipher, the decryption function is determined by solving the encryption function for m in terms of c; that is, $m \equiv c - k \equiv c + (26 - k) \pmod{26}$, where the decryption key is $d = 26 - k \in \mathbb{Z}_{26}$. Given the ciphertext from the previous example, the first step of decryption is to convert the ciphertext to its numerical value, c, using Table 2. Then the function $m \equiv c - 3 \pmod{26} \equiv c + 23 \pmod{26}$ is used to obtain the plaintext as shown in the table below:

ciphertext	*p*	*d*	*w*	*k*
c	15	3	22	10
$m \equiv c - 3 \pmod{26}$	$15 - 3 \equiv 12$	$3 - 3 \equiv 0$	$22 - 3 \equiv 19$	$10 - 3 \equiv 7$
plaintext	M	A	T	H

The problem with the security of the cipher is that it can be solved by either frequency analysis or a brute force attack. Frequency analysis is using the frequency that each ciphertext letter appears and comparing that to the frequency that English letters are used in words and sentences [2]. The most common letters, in order, are E, T, A, O, I, N, and S. A frequency table for all letters can be found in [2]. A brute force attack is trying all different 25 keys until the correct one is found. This was time consuming in the time period of Julius Caesar, but with technology now, it can be accomplished in seconds.

3 Vigenère Cipher

3.1 Polyalphabetic Cipher

The next major type of cipher that is analyzed is the polyalphabetic cipher. This type of cipher is similar to a monoalphabetic cipher; however, unlike a monoalphabetic cipher, a polyalphabetic cipher has more than one cipher alphabet [2]. What this means is, compared to Table 1, a polyalphabetic cipher would have two or more rows of shifted or rearranged letters depending on the key. The use of the different cipher alphabets varies but one example is the first cipher alphabet is used for even spaced letters while the second cipher alphabet is used for odd spaced letters. This would mean the encryption alternates between the two ciphers. One of the most famous polyalphabetic ciphers is the Vigenère cipher.

3.2 Introduction

Blaise de Vigenère (1523-1596) was a French diplomat and cryptographer who did not explicitly state that he created the following cipher, but is accredited with creating it [4][1]. The Vigenère Keyword cipher, usually known just as the Vigenère cipher, is a polyalphabetic cipher that uses one or more keywords or letters as the key.

3.3 Encryption

A description of the Vigenère cipher is as follows:
Let $n \in \mathbb{Z}^+$ with $n \geq 2$. The key of the Vigenère cipher is a keyword or phrase, given by $k_1, k_2, ..., k_n$ where each k_i is the numerical value of each letter. In order to encrypt the message, the message is split into blocks of length n. Note that this is the length of the keyword. The message is converted to its numerical equivalent, $m_1, m_2, ..., m_n$, using Table 2. Then m_i is converted to the numerical value, $c_i \in \mathbb{Z}_{26}$, of the ciphertext using the following encryption function:

$$c_i \equiv m_i + k_i \pmod{26}.$$

Lastly, c_i is converted back to letters, as the ciphertext, using Table 2.
To illustrate the Vigenère cipher, the plaintext "math is fun" is encrypted with keyword *jim*, which is represented as (9, 8, 12).

plaintext	m	a	t	h	i	s	f	u	n
m_i	12	0	19	7	8	18	5	20	13
keyword	j	i	m	j	i	m	j	i	m
k_i	9	8	12	9	8	12	9	8	12
$c_i \equiv m_i + k_i \pmod{26}$	21	8	5	16	16	4	14	2	25
ciphertext	v	i	f	q	q	e	o	c	z

Thus the ciphertext *vifqqeocz* is achieved.

3.4 Decryption

For the Vigenère cipher, the decryption process is determined by solving each encryption function for m_i in terms of c_i; that is, $m_i \equiv c_i - k_i \equiv c_i + (26 - k_i) \pmod{26}$, where the decryption key,

d_i, is given by $d_i = 26 - k_i$ with $k_i \in \mathbb{Z}_{26}$ for every $i \in \{1, 2, ..., n\}$.

Suppose someone wanted to decrypt the previous example's ciphertext. Then, the ciphertext *vifqqeocz* will be decrypted with the decryption key $(26 - 9, 26 - 8, 26 - 12) = (17, 18, 14)$. The decryption process is shown in the following table:

ciphertext	v	i	f	q	q	e	o	c	z
c_i	21	8	5	16	16	4	14	2	25
d_i	17	18	14	17	18	14	17	18	14
$m_i \equiv c_i + d_i \pmod{26}$	12	0	19	7	8	18	5	20	13
plaintext	m	a	t	h	i	s	f	u	n

Thus the plaintext "math is fun" is achieved.

Despite being more secure than the monoalphabetic ciphers at the time of its creation (1586), the more complex nature of the encryption process made the Vigenère cipher unpopular for its time. It randomly resurfaced in the late 1700's when it was used a little by cipher secretaries. Then the Vigenère and other polyalphabetic ciphers were strongly used around the early nineteenth century with telegraphs. Then, in the late 1800's it was cracked by Friedrich Wilhelm Kasiski and was deemed no longer secure [1]. While the Vigenère cipher is more secure than monoalphabetic ciphers, it is still very insecure, especially with today's technology. The secrecy of the key and the key length is what makes the Vigenère cipher secure. If the key length was able to be found, it would then be susceptible to the same attacks as monoalphabetic shift ciphers, that being frequency analysis and brute force attacks. There are two tests that are used to estimate the key length of a Vigenère cipher. These two tests are the Friedman test and the Kasiski test [2]. These test are described in detail in [2]. After the key length is found or estimated, the ciphertext is broken up into sections of length equal to the key length. Then each section is essentially like a monoalphabetic cipher and can be broken using frequency analysis.

4 Block Ciphers

4.1 Introduction

Block ciphers are ciphers that encrypt blocks of plaintext to blocks of ciphertext, instead of one letter or number at a time [3]. Cryptosystems such as the Playfair cipher, Advanced Encryption

Standard (AES), and Hill cipher are block ciphers [2]. The Hill cipher is significant because it was most likely the first cipher that major mathematical ideas (modular arithmetic and linear algebra) were used in cryptography [3]. In 1929, a mathematician by the name of Lester Hill described his cipher in an article called *Cryptography in an Algebraic Alphabet* [2].

4.2 Encryption

To use the Hill cipher, begin by selecting an encryption key which is an $n \times n$ matrix, A. Note that if the ciphertext is to be decrypted, $A^{-1} \pmod{26}$ must exist so the determinant of A must satisfy

$$\text{gcd}(\det(A), 26) = 1.$$

Assign each plaintext letter a numerical value using Table 2. Break the plaintext of length i into $1 \times n$ matrices. If the last plaintext number, m_i, does not fill up the last matrix, add an x (a numerical value of 23) to the matrix until it is full. Multiply each $1 \times n$ matrix by A and take the matrix modulo 26 to get the ciphertext, c_i, with matrices of length $1 \times n$. The encryption process for $n = 2$ is modeled below:

$$\begin{aligned}
\begin{bmatrix} c_1 & c_2 \end{bmatrix} &\equiv \begin{bmatrix} m_1 & m_2 \end{bmatrix} A \pmod{26}, \\
\begin{bmatrix} c_3 & c_4 \end{bmatrix} &\equiv \begin{bmatrix} m_3 & m_4 \end{bmatrix} A \pmod{26}, \\
&\vdots \\
\begin{bmatrix} c_{i-1} & c_i \end{bmatrix} &\equiv \begin{bmatrix} m_{i-1} & m_i \end{bmatrix} A \pmod{26}.
\end{aligned}$$

The numerical values $c_1, c_2, c_3, ..., c_i$ are then reverted back into letters to get the ciphertext.

The encryption process will be demonstrated by encrypting "math is fun" using the Hill cipher with an encryption key $A = \begin{bmatrix} -4 & 13 \\ 11 & 5 \end{bmatrix}$.

Using Table 2, "Math is fun" is assigned numerical values and broken up into sets of 1×2 matrices

$$\begin{bmatrix} 12 & 0 \end{bmatrix}, \begin{bmatrix} 19 & 7 \end{bmatrix}, \begin{bmatrix} 8 & 18 \end{bmatrix}, \begin{bmatrix} 5 & 20 \end{bmatrix}, \begin{bmatrix} 13 & 23 \end{bmatrix}.$$

Note that an x, with value of 23, was added to the last matrix because the last matrix was not filled. Then each matrix is multiplied by A (mod 26):

$$\text{m and a } \Rightarrow \begin{bmatrix} 12 & 0 \end{bmatrix} \begin{bmatrix} -4 & 13 \\ 11 & 5 \end{bmatrix} \pmod{26} \equiv \begin{bmatrix} 4 & 0 \end{bmatrix} \pmod{26} \Rightarrow e \text{ and } a,$$

$$\text{t and h } \Rightarrow \begin{bmatrix} 19 & 7 \end{bmatrix} \begin{bmatrix} -4 & 13 \\ 11 & 5 \end{bmatrix} \pmod{26} \equiv \begin{bmatrix} 1 & 22 \end{bmatrix} \pmod{26} \Rightarrow b \text{ and } w,$$

$$\text{i and s } \Rightarrow \begin{bmatrix} 8 & 18 \end{bmatrix} \begin{bmatrix} -4 & 13 \\ 11 & 5 \end{bmatrix} \pmod{26} \equiv \begin{bmatrix} 10 & 12 \end{bmatrix} \pmod{26} \Rightarrow k \text{ and } m,$$

$$\text{f and u } \Rightarrow \begin{bmatrix} 5 & 20 \end{bmatrix} \begin{bmatrix} -4 & 13 \\ 11 & 5 \end{bmatrix} \pmod{26} \equiv \begin{bmatrix} 18 & 9 \end{bmatrix} \pmod{26} \Rightarrow s \text{ and } j,$$

$$\text{n and x } \Rightarrow \begin{bmatrix} 13 & 23 \end{bmatrix} \begin{bmatrix} -4 & 13 \\ 11 & 5 \end{bmatrix} \pmod{26} \equiv \begin{bmatrix} 19 & 24 \end{bmatrix} \pmod{26} \Rightarrow t \text{ and } y.$$

Thus the ciphertext is *eabwkmsjty*.

4.3 Decryption

In a Hill cipher, the decryption key is given by $A^{-1} \pmod{26}$. In a similar fashion as encryption, break the ciphertext with numerical values into $1 \times n$ matrices. Then multiply each matrix by $A^{-1} \pmod{26}$ to determine the plaintext. The decryption process for $n = 2$ is modeled below:

$$\begin{bmatrix} m_1 & m_2 \end{bmatrix} \equiv \begin{bmatrix} c_1 & c_2 \end{bmatrix} A^{-1} \pmod{26},$$

$$\begin{bmatrix} m_3 & m_4 \end{bmatrix} \equiv \begin{bmatrix} c_3 & c_4 \end{bmatrix} A^{-1} \pmod{26},$$

$$\vdots$$

$$\begin{bmatrix} m_{i-1} & m_i \end{bmatrix} \equiv \begin{bmatrix} c_{i-1} & c_i \end{bmatrix} A^{-1} \pmod{26}.$$

The numerical values are reverted back to letters to reveal the plaintext.

As an example, suppose the matrix $A = \begin{bmatrix} -4 & 13 \\ 11 & 5 \end{bmatrix}$ is the encryption key. Then $A^{-1} \pmod{26}$

is determined as follows:

$$\begin{aligned}
A^{-1} &\equiv (\det A)^{-1} \begin{bmatrix} 5 & -13 \\ -11 & -4 \end{bmatrix} \pmod{26} \\
&\equiv (-4(5) - 13(11))^{-1} \begin{bmatrix} 5 & -13 \\ -11 & -4 \end{bmatrix} \pmod{26} \\
&\equiv (-163)^{-1} \begin{bmatrix} 5 & -13 \\ -11 & -4 \end{bmatrix} \pmod{26} \\
&\equiv 19^{-1} \begin{bmatrix} 5 & -13 \\ -11 & -4 \end{bmatrix} \pmod{26} \\
&\equiv 11 \begin{bmatrix} 5 & -13 \\ -11 & -4 \end{bmatrix} \pmod{26} \\
&\equiv \begin{bmatrix} 55 & -143 \\ -121 & -44 \end{bmatrix} \pmod{26} \\
&\equiv \begin{bmatrix} 3 & 13 \\ 9 & 8 \end{bmatrix} \pmod{26}.
\end{aligned}$$

Thus, $\begin{bmatrix} 3 & 13 \\ 9 & 8 \end{bmatrix}$ is the decryption matrix. The decryption of *eabw* is achieved as follows:

$$e \text{ and } a \Rightarrow \begin{bmatrix} 4 & 0 \end{bmatrix} \begin{bmatrix} 3 & 13 \\ 9 & 8 \end{bmatrix} \pmod{26} \equiv \begin{bmatrix} 12 & 0 \end{bmatrix} \Rightarrow \text{ m and a}$$

$$b \text{ and } w \Rightarrow \begin{bmatrix} 1 & 22 \end{bmatrix} \begin{bmatrix} 3 & 13 \\ 9 & 8 \end{bmatrix} \pmod{26} \equiv \begin{bmatrix} 19 & 7 \end{bmatrix} \Rightarrow \text{ t and h}$$

Thus, *eabw* decrypts to “math.”

The Hill cipher was not used because it was only marginally more secure than other ciphers at the time, and the lack of technology made the cipher tedious to use [2].

5 Public Key Ciphers

5.1 Introduction

A feature the previous ciphers have in common is the fact that the key is discussed ahead of time. There is no way for the recipient to know the key if the two parties have never met before and with increased reliance on computer technology, secure communication of keys was becoming more of an issue. By the late 1970's, "the cost and delay imposed by this key distribution problem [was] a major barrier to the transfer of business communications to large teleprocessing networks" [5]. In 1976 an idea on how to combat this problem was described in the paper *New Directions in Cryptography*, by Whitfield Diffie and Martin Hellman [5]. A public key cryptosystem, also called asymmetric key cryptosystem, has two keys, a public encryption key and a private decryption key. It is called a public key cryptosystem because everyone has access to the encryption key, not just the sender and recipient. While everyone has access to the encryption key, no one besides the recipient has the decryption key, and it is beyond computational abilities to get the decryption key from the encryption key; herein lies the security of public key cryptosystems [4]. A non-mathematical way to think about public key cryptosystems follows [3].

Example: Bob sends Alice a box and an unlocked padlock. Alice puts her message in the box, locks Bob's lock on it, and sends the box back to Bob. Once Bob receives the box back, Bob, and only Bob, can open the box and read the message, since he is the only person with the key.

Diffie and Hellman described a public key cryptosystem in their paper [5]. The following is a simplification of this:

	Public Encryption Key	Private Decryption Key
Alice	E_a	D_a
Bob	E_b	D_b

Encryption:

Suppose that Alice wants to send a message, M, to Bob. Alice first looks up Bob's public encryption key, E_b, and uses it to encrypt M; that is, $C = E_b(M)$. Then Alice sends the encrypted message, C, to Bob.

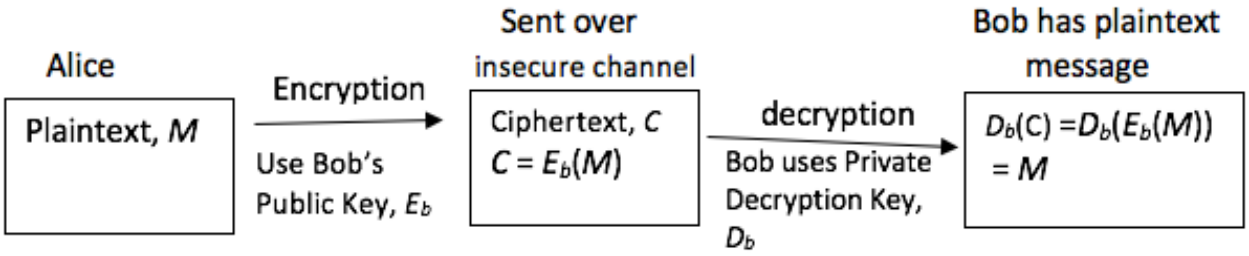
Decryption:

In order to decrypt C, Bob uses his private decryption key, D_b, as follows:

$$D_b(C) = D_b(E_b(M)) = M,$$

since D_b and E_b are inverse functions of each other.

The following is a visual representation of this process:



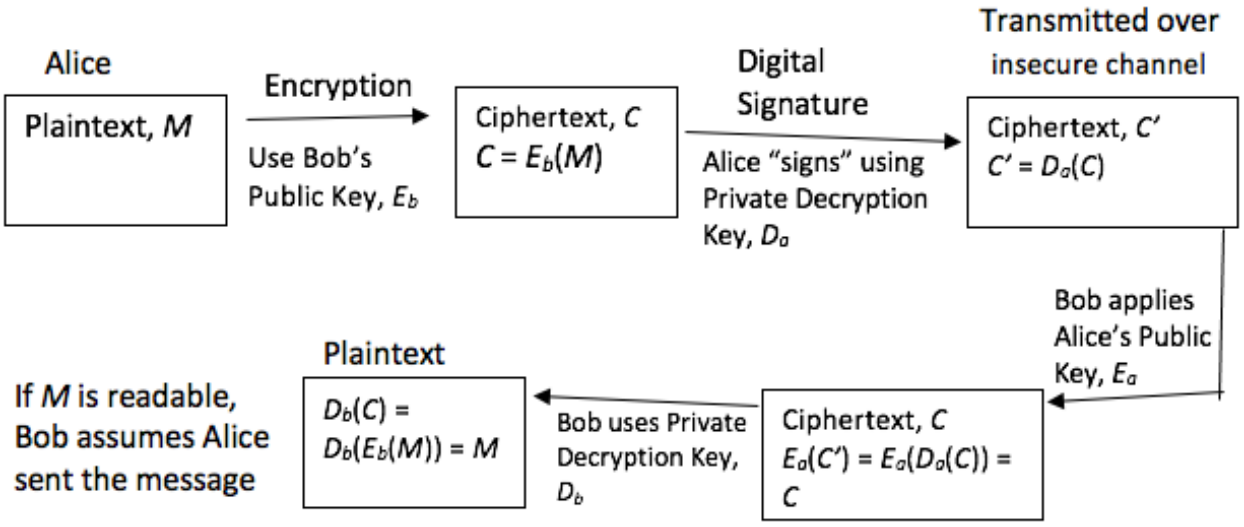
Next consider how messages are authenticated in a public key system. Everyone has access to the public key and could use it to pretend to be someone else. So, when Bob receives a message from Alice, how does Bob know that the message was actually from Alice? This is message authentication. The way to properly certify a personal signature is to encrypt a message with a digital signature, which is a method for authenticating ("signing") a message to verify it was sent by the specified person.

In a public key cryptosystem, the digital signature can be built into the encryption process using the public key algorithm. This is further expanded upon in section **5.5**.

To begin, Alice follows the public key encryption process to get the ciphertext, C. Then, Alice adds her signature to the message, C, with her private key, D_a, which yields
$C' = D_a(C) = D_a(E_b(M))$. Then Alice sends C' to Bob.

In order for Bob to recover the plaintext message, M, he must first apply Alice's public key, E_a, which yields, $E_a(C') = E_a(D_a(C)) = C$. Next, he applies his private decryption key, D_b, which yield, $D_b(C) = D_b(E_b(M)) = M$. If the plaintext message M is readable, then Bob is confident that Alice sent the message.

A visual summary of the authentication process is as follows:



5.1.1 Mathematical Introduction

While the concept of a public key cipher was nice in theory, Diffie and Hellman did not provide a practical example of a public-key cryptosystem. A year after their paper was published, in 1977, three researchers from MIT Laboratory for Computer Science answered that question with the RSA cryptosystem in [1]. The RSA cryptosystem is named after the three MIT researchers to publish the cryptosystem, Ronald Rivest, Adi Shamir and Leonard Adleman [1]. The number theory applications of prime numbers, modular exponentiation, and Euler's Theorem are integral to successfully implement the RSA cryptosystem.

The following definitions and theorems are necessary in explaining how the RSA cryptosystem works.

Definition 1. (Prime Number). *An integer $p > 1$ is a prime number if and only if the only positive integers to divide p are itself and 1.*

Definition 2. (Relatively Prime). *Two positive integers m,n are relatively prime if and only if gcd(m,n)=1.*

Definition 3. (Euler's Phi Function). *Let n be a positive integer. The number of positive integers less than or equal to n that are relatively prime to n, denoted $\phi(n)$, is* ***Euler's Phi Function****.*

Note that the integers $1, 2, 3, ..., p-1$ are each relatively prime to the prime. Thus, by definition of Euler's Phi Function, $\phi(p) = p - 1$. We state this result in the following theorem.

Theorem 1. *If p is prime, then* $\phi(p) = p - 1$.

For the purpose of the RSA cryptosystem, the recipient will need to compute $\phi(N)$ where $N = pq$ and p and q are distinct primes. Then, $\phi(N)$ is calculated as follows.
Note that p, q are positive integers less than N. Since the relatively prime integers are the only integers needed, the multiples of p are discarded, namely $p, 2p, 3p, ..., qp$; that is, the q multiples of p are discarded. Similarly, the p multiples of q are discarded, $q, 2q, 3q, ..., pq$. However, pq has been discarded twice, so pq is added back one time to be counted as a relatively prime integer less than or equal to pq. Thus,

$$\phi(N) = \phi(pq) = pq - q - p + 1 \quad = q(p-1) - (p-1) = (p-1)(q-1) \quad = \phi(p)\phi(q).$$

We state this result in the following theorem.

Theorem 2. *If p and q are distinct primes, then* $\phi(pq) = \phi(p)\phi(q) = (p-1)(q-1)$.

The following theorem is also needed in explaining why decryption works in RSA. A proof of this theorem can be found in [4].

Theorem 3. (Euler's Theorem). *If* $n \in \mathbb{Z}^+$ *and* $a \in \mathbb{Z}$ *such that a and n are relatively prime, then* $a^{\phi(n)} \equiv 1$ *(*mod *n).*

The following is an example of Euler's Theorem. Take $a = 15$ and $n = 11(17) = 187$. Note that $\phi(187) = \phi(11)\phi(17) = 10(16) = 160$, and $\gcd(15, 187) = 1$. Thus, by Euler's Theorem, $15^{\phi(187)} \equiv 15^{160} \equiv 1 \pmod{187}$.

5.2 RSA

Unlike symmetric key cryptosystems, the initiation of the RSA cipher begins with the recipient. The first step for the recipient in using the RSA cryptosystem is to create a number *N* such that $N = pq$ where *p* and *q* are distinct prime numbers. The larger the *p* and *q*, the more secure the cipher is, and in fact with today's technology, primes *p* and *q* need to each be around 100 digits long and slightly different lengths of digits long [3]. Next, the recipient calculates $\phi(N) = (p-1)(q-1)$ and then selects an integer *e* such that $1 < e < \phi(N)$ and $\gcd(e, \phi(N)) = 1$. The purpose of selecting *e* in this manner is to ensure that $e^{-1} \pmod{\phi(N)}$ exists. The number *N* is called the

enciphering modulus, while the number *e* is called the enciphering exponent [4]. The enciphering modulus and exponent make up the public key for the cipher, as the public key (N, e) is made public, (for example, in a directory), so any person who desires can send the recipient a message. The last step for the recipient is to calculate $d \equiv e^{-1} \pmod{\phi(N)}$. This means finding $d \in \mathbb{Z}^+$ and $1 < d < \phi(N)$ such that $de \equiv 1 \pmod{\phi(N)}$. This number *d* is called the decryption exponent [3]. The decryption exponent is kept private along with *p* and *q*. Thus, (*p,q,d*) is the private key and is only known by the recipient.

The first step for the sender is to convert the plaintext message to numbers using the ASCII table. The ASCII, or American Standard Code for Information Interchange, is a table of corresponding characters, numbers, and letters [2]. It is similar to Table 2, but it starts at 32 as 0-31 is reserved for control characters [2]. The version of the ASCII table from [2] will be used for the rest of the paper and can be found at the end of the paper in the "Diagrams" section.

After the sender converts the plaintext to numbers, m, the following congruence is used to convert m to the ciphertext, c, and is sent to the recipient.

$$c \equiv m^e \pmod{N}.$$

Lastly, the recipient of the ciphertext converts c to m by computing $m \equiv c^d \pmod{N}$. Then, m is converted back into letters using the ASCII table. This the basic explanation of the RSA cryptosystem. The following will show that $c^d \pmod{N}$ yields the original message, m. Observe on the following page:

$$\begin{aligned}
c^d \pmod{N} &\equiv (m^e)^d \pmod{N} \\
&\equiv m^{ed} \pmod{N} \quad [\text{by note 1 below}] \\
&\equiv m^{1+\phi(N)k} \pmod{N} \\
&\equiv m^1 \cdot m^{\phi(N)k} \pmod{N} \\
&\equiv m^1 \cdot (m^{\phi(N)})^k \pmod{N} \quad [\text{by note 2 below}] \\
&\equiv m(1)^k \pmod{N} \\
&\equiv m \pmod{N}
\end{aligned}$$

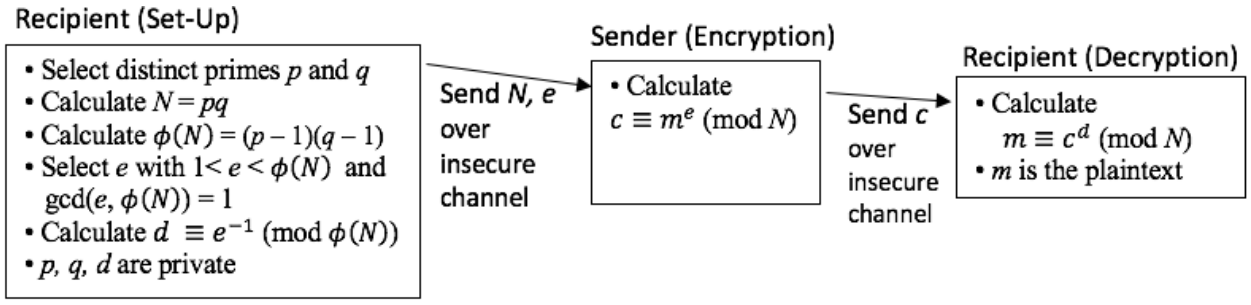
Note 1: Since e and d are inverses modulo $\phi(N)$, we deduce that,

$$ed \equiv 1 (\text{mod } \phi(N)) \Rightarrow ed = 1 + \phi(N)k, \quad \text{for some } k \in \mathbb{Z}.$$

Note 2: By Theorem 3, Euler's Theorem, $m^{\phi(N)} \equiv 1 \pmod{N}$ provided $\gcd(m, N) = 1$.

Note: Most likely m and N are relatively prime since in implementation of RSA, N is the product of two large primes.

We summarize the RSA encryption and decryption in the diagram below.



The following is an example to illustrate how RSA encryption and decryption works.

The sender wants to encrypt and send the message "Math is fun" to the recipient. Let $p = 605837$ and $q = 40605347$. Then $N = 24600221610439$ and $\phi(N) = 24600180399256$. Let $e = 3752137$. Note that $\gcd(24600180399256, 3752137) = 1$. Then observe that $d \equiv 3752137^{-1} \pmod{24600180399256}$, thus $d = 14512599598401$. Then the recipient sends $(24600221610439, 3752137)$ to the sender.

The sender then turns the phrase "Math is fun!" into blocks of numerical values, m_i, using the ASCII table. "Math" $= 7797116104 = m_1$, " is " $= 3210511532 = m_2$ (note that the space before and after "is" is included), "fun!" $= 10211711033 = m_3$. Then observe the following encryption of the plaintext, m, to ciphertext, c.

$$c_1 = 3427489827140 \equiv 7797116104^{3752137} \pmod{24600221610439}$$

$$c_2 = 6257661437078 \equiv 3210511532^{3752137} \pmod{24600221610439}$$

$$c_3 = 18554069556725 \equiv 10211711033^{3752137} \pmod{24600221610439}$$

Then the values of c_1, c_2, c_3 are sent to the recipient.

The recipient then takes the values of c_1, c_2, c_3 and decrypts them as follows:

$$m_1 = 7797116104 \equiv 3427489827140^{14512599598401} \pmod{24600221610439}$$

$$m_2 = 3210511532 \equiv 6257661437078^{14512599598401} \pmod{24600221610439}$$

$$m_3 = 10211711033 \equiv 18554069556725^{14512599598401} \pmod{24600221610439}$$

Then, using the ASCII table, the numerical values of m are transformed back into the plaintext "Math is fun!".

5.3 Diffie-Hellman Key Exchange

Before writing their paper in 1976, Martin Hellman and Whitfield Diffie tried to solve the problem of key distribution, or how to get the key in a symmetric key cryptosystem from the sender to the recipient. Then they were joined by Ralph Merkle and the three of them attacked the key distribution problem [1]. The problem was trying to get the key from the sender to the recipient without the key being discovered by a third party. If the key was encrypted and sent, the recipient would have to know a second key to decrypt the encryption of the first key and so on. If the keys could not be shared in person, how would they get shared secretly between the two parties? The answer to this question is a one-way function based on modular arithmetic and exponentiation. A one-way function is a function that is easy to compute forwards, but extremely hard to compute backwards, while a two-way function is a function that is easy to compute forwards and backward [1]. An example of a two-way function is a linear function such as $f(x) = 2x$. It is easy to undo the function to find that $f^{-1}(x) = \frac{x}{2}$. However, with one-way functions, such as modular exponentiation, multiplying and factoring, and discrete logarithms, it is almost impossible to reverse them. After years of contemplation, Hellman, Diffie, and Merkle discovered the Diffie-Hellman key exchange (though some refer to it as the Diffie-Hellman-Merkle key exchange) [1]. The three researchers "publicly demonstrated their discovery at the National Computer Conference in June 1976" [1]. Then Diffie and Hellman published their findings, along with some other ideas in their paper *New Directions in Cryptography* [5].

5.3.1 Steps

There is a way to incorporate the Diffie-Hellman key exchange into the RSA encryption process. The Diffie-Hellman key exchange makes sure that the enciphering exponent, *e*, is secret and secure. The first step of this process is the same as in using the RSA cryptosystem.

First the recipient calculates *N* such that $N = pq$ where *p* and *q* are distinct prime numbers. Then $\phi(N)$ is calculated. Then the process differs. Next, an integer *k* is chosen by the recipient such that $\gcd(k, N) = 1$ and $1 < k < N$. Then the recipient calculates $\ell \equiv k^r \pmod{N}$ with $r \in \mathbb{Z}^+$ and $1 < r < N$. The recipient makes ℓ, k and N public to be accessed by the sender, keeping the value of r private.

The sender follows similar steps. The sender selects an $s \in \mathbb{Z}^+$ such that $1 < s < N$ and calculates $z \equiv k^s \pmod{N}$. The sender then makes the value z public for the recipient and keeps s private.

For both the recipient and the sender to obtain the enciphering exponent, e, to be used in the RSA cryptosystem, the recipient calculates $z^r \pmod{N}$ while the sender calculates $\ell^s \pmod{N}$. Note that the result will yield the same enciphering exponent e as shown below.

$$\text{Recipient: } z^r \equiv (k^s)^r \equiv k^{sr} \equiv k^{rs} \pmod{N}$$

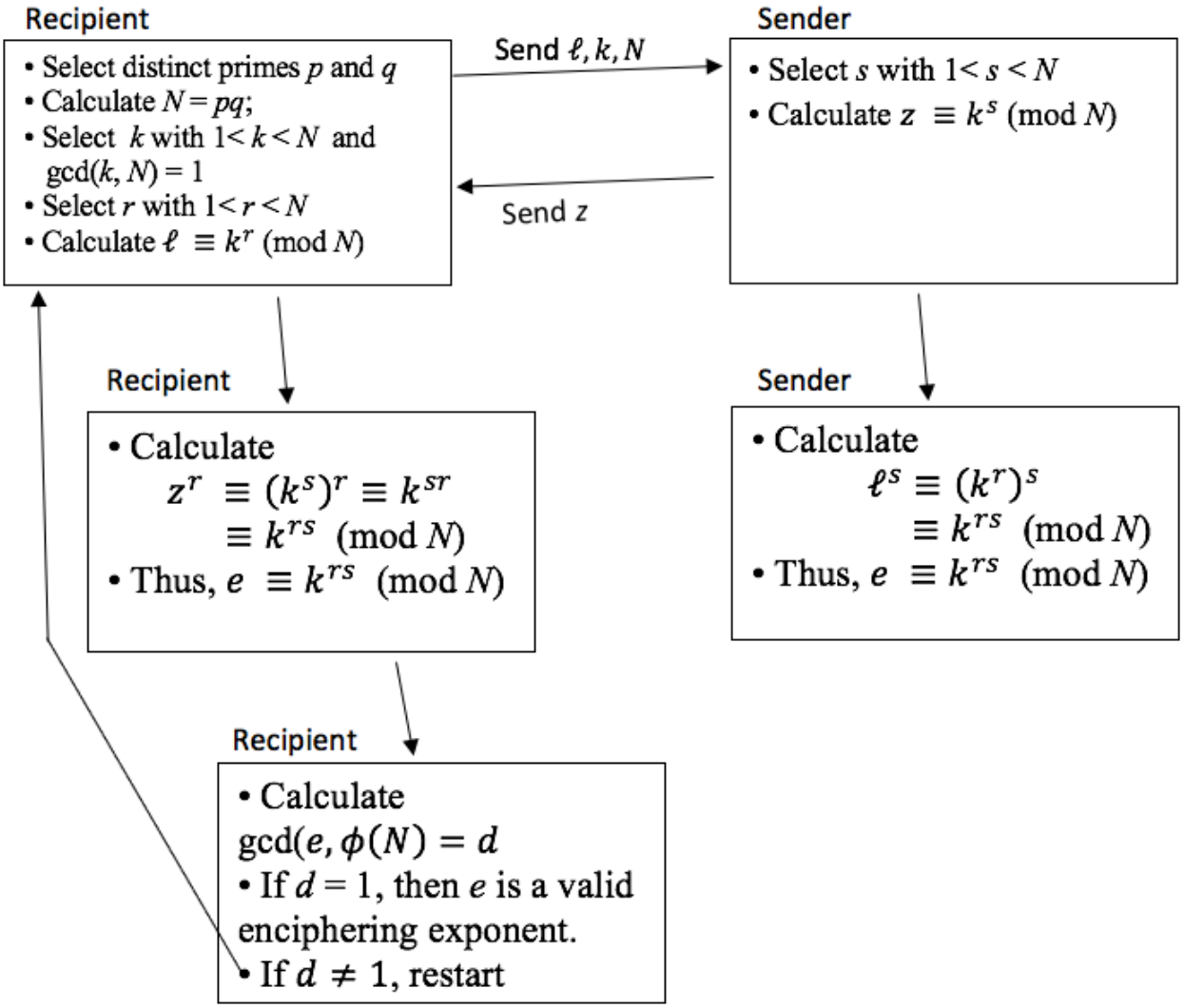
$$\text{Sender: } \ell^s \equiv (k^r)^s \equiv k^{rs} \pmod{N}$$

Thus, for both the recipient and sender, $e \equiv k^{rs} \pmod{N}$.

Recall in the RSA cryptosystem that the enciphering exponent, e, must be selected so that $\gcd(e, \phi(N)) = 1$. The recipient checks to determine if e is an acceptable enciphering exponent. If $\gcd(e, \phi(N)) \neq 1$, then the above process is repeated with the recipient selecting different values for k or r. The process continues until an enciphering exponent is generated that satisfies $\gcd(e, \phi(N)) = 1$.

After a usable enciphering exponent is chosen, the RSA cryptosystem process continues as normal starting from calculating the decryption exponent.

Although the process above is being used to privatize the enciphering exponent in the RSA process, the steps above will work for any cipher where, after e is calculated, e is the key. A summary of the key exchange is shown in the diagram below.



The following is an example to illustrate how the Diffie-Hellman key exchange works.

Let $p = 375109127$ and $q = 555404209$. Then $N = 208337187970115543$. Let $k = 4589423$. Note that $\gcd(4589423, 208337187970115543) = 1$. Let $r = 127854931$. Then observe that $\ell \equiv 4589423^{127854931} \pmod{208337187970115543}$; thus, $\ell = 221453102918875$. The recipient then sends $(221453102918875, 4589423, 208337187970115543)$ to the sender.

Let $s = 328567$. Then observe that $z \equiv 4589423^{328567}$ (mod208337187970115543); thus,

$z = 139663417434917067$. Then 139663417434917067 is send to the recipient. Then observe that for the recipient,

$e \equiv 139663417434917067^{127854931} \equiv 4589423^{127854931 \cdot 328567} (\text{mod } 208337187970115543)$;

thus, $e = 45312272687948005$. Then observe that for the sender,

$e \equiv 221453102918875^{328567} \equiv 4589423^{127854931 \cdot 328567} (\text{mod } 208337187970115543)$;

thus, $e = 45312272687948005$. Let $\phi(208337187970115543) = 208337187039602208$. Note that $\gcd(45312272687948005, 208337187039602208) = 1$. Thus, $e = 45312272687948005$ is a valid enciphering exponent.

5.3.2 Discrete Logarithm

While incorporating the Diffie-Hellman key exchange into the RSA cryptosystem, suppose an eavesdropper got the following information, $z, \ell, N,$ and k, since these were made public. From this information, an eavesdropper would want to find the enciphering exponent e. In order to find e, the eavesdropper would need to determine r from the congruence $\ell \equiv k^r \pmod{N}$ and s from the congruence $z \equiv k^s \pmod{N}$.

The integer r is called the discrete logarithm of ℓ to the base k modulo N. Similarly, s is the discrete logarithm of z to the base k mod N. While modular exponentiation is fairly easy, the reverse process of finding the discrete logarithm is challenging. This is the discrete logarithm problem.

The discrete logarithm problem, in general, is trying to find an integer x that satisfies the congruence $\beta \equiv \alpha^x \pmod{p}$ where p is prime and α is a primitive root modulo p. The exponent x is called the discrete logarithm of β with base α mod p and is denoted by $x = L_\alpha(\beta)$.

For example, in the congruence $13^r \equiv 17 \pmod{479}$, r is the discrete logarithm of 17 with base 13 modulo 479; that is, $r = L_{13}(17)$. In this case, $r = 237$.

5.4 ElGamal

The next advancement in the use of number theory in cryptography was the ElGamal cryptosystem. This cryptosystem was created and published in 1985 by Taher Elgamal [6]. The security of the ElGamal cryptosystem comes from the difficulty of computing discrete logarithms [6]. The initial step in implementing the ElGamal cryptosystem is for the recipient to choose a large prime number p and an integer g that is a primitive root mod p and $1 \leq g < \phi(p)$. Then the recipient chooses an integer r such that $1 \leq r < \phi(p)$ and calculates $h \equiv g^r \pmod{p}$. Selecting g as a primitive root mod p guarantees that h will have a well-defined discrete logarithm. The values of $p, g,$ and h are made public for the sender, but r is kept private.

Once the recipient has completed the initial set-up of the ElGamal cryptosystem, the sender, upon receiving the values of $p, g,$ and h, encrypts the plaintext message as follows. The sender first

converts the plaintext message into a numerical value, m, by using the ASCII table. The sender then selects $s \in \mathbb{Z}^+$ with $1 < s < \phi(p) = p - 1$, and calculates $\ell \equiv g^s \pmod{p}$. The sender encrypts m into the ciphertext c by calculating $c \equiv m \cdot h^s \pmod{p}$. For increased security, s should be random and change for each encryption. Finally, ℓ and c are sent to the recipient by the sender.

Once the recipient receives ℓ and c, the recipient decrypts the ciphertext, c, and obtains the plaintext value, m, by calculating $m \equiv c \cdot \ell^{-r} \pmod{p}$. This congruence yields m as follows:

$$\begin{aligned} c \cdot \ell^{-r} &\equiv (mh^s)(g^s)^{-r} \pmod{p} \quad \text{[by Note 1 below]} \\ &\equiv m(g^r)^s(g^s)^{-r} \pmod{p} \quad \text{[by Note 2 below]} \\ &\equiv m(g^{rs})(g^{-rs}) \pmod{p} \\ &\equiv m(g^0) \pmod{p} \\ &\equiv m \pmod{p}. \end{aligned}$$

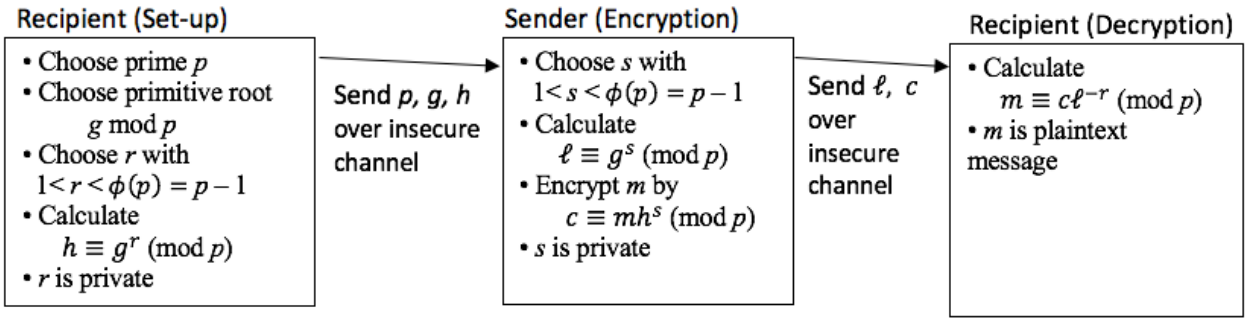
Note 1: Recall that $c \equiv mh^s \pmod{p}$ and $\ell \equiv g^s \pmod{p}$.

Note 2: Recall that $h \equiv g^r \pmod{p}$.

Consequently, $m \equiv c\ell^{-r} \pmod{p}$. Lastly, m is converted back into characters by the ASCII table.

The security of the ElGamal cryptosystem lies in trying to find r, which is private. Finding r would require solving the congruence $h \equiv g^r \pmod{p}$, which is a discrete logarithm problem.

We summarize the ElGamal cryptosystem in the following diagram.



The following is an example:

The sender wants to encrypt and send the message "Math is fun!" to the recipient.

Let $p = 738733242911497$, $g = 13$, and $r = 45691$. Then observe that

$h \equiv 13^{45691} \pmod{738733242911497}$; thus, $h = 175778470844015$. Then the recipient sends $(738733242911497, 13, 175778470844015)$ to the sender. Then let $s = 607512$. Then observe that $\ell \equiv 13^{607512} \pmod{738733242911497}$; thus, $\ell = 348425674930505$. The sender then turns the phrase "Math is fun!" into blocks of numerical values, m_i, using the ASCII table.
"Math" $= 7797116104 = m_1$, " is " $= 3210511532 = m_2$ (note that the space before and after "is" is included), "fun!" $= 10211711033 = m_3$. Then observe the following encryption of the plaintext, m, to ciphertext, c.

$$c_1 = 15303114233367 \equiv 7797116104 \cdot 175778470844015^{607512} \pmod{738733242911497}$$
$$c_2 = 110496918609746 \equiv 3210511532 \cdot 175778470844015^{607512} \pmod{738733242911497}$$
$$c_3 = 372322954090376 \equiv 10211711033 \cdot 175778470844015^{607512} \pmod{738733242911497}$$

Then the values of c_1, c_2, c_3 are sent to the recipient.
The recipient then takes those values and decrypts them as follows:

$$m_1 = 7797116104 \equiv 15303114233367 \cdot 348425674930505^{-45691} \pmod{738733242911497}$$
$$m_2 = 3210511532 \equiv 110496918609746 \cdot 348425674930505^{-45691} \pmod{738733242911497}$$
$$m_3 = 10211711033 \equiv 372322954090376 \cdot 348425674930505^{-45691} \pmod{738733242911497}$$

Then using the ASCII table, the numerical values of m are transformed back into the plaintext "Math is fun!".

5.5 Digital Signatures

Since everyone has access to the public key that the recipient puts out, anyone can send the recipient a message. This can become a problem when an imposter sends the recipient a message claiming to be someone else. This problem is solved with digital signatures. First we discuss the RSA digital signature which is described in great detail by Rivest, Shamir and Adleman in their paper *A Method for Obtaining Digital Signitures and Public-Key Cryptosystems* [7].

5.5.1 RSA

The first step begins as the RSA cryptosystem. The recipient selects two large distinct primes p, q and computes $N = pq$. Then, an $e_0 \in \mathbb{Z}^+$ with $1 < e_0 < \phi(N)$ such that gcd($e_0, \phi(N)$) is chosen. Next, the recipient calculates a d_0 such that $e_0 d_0 \equiv 1 \pmod{\phi(N)}$. The recipient's signature is given by the following congruence: $s \equiv m^{d_0} \pmod{N}$. Then the recipient makes (e_0, N) and (m, s) public.

The sender of the message calculates t using $t \equiv s^{e_0} \pmod{N}$. If $t = m$, then the signature is valid. Showing that $m \equiv s^{e_0} \pmod{N}$ is identical to showing how RSA decryption works as shown on page 14.

5.5.2 ElGamal

The digital signature for the ElGamal cryptosystem starts the same as well. After (p, g, h) are made public is where the digital signature steps begin. First the recipient selects a secret k such that gcd($k, \phi(p)$) $= 1$. Then the recipient calculates two values α, β using the following congruences:

$$\alpha \equiv g^k \pmod{p} \text{ (with } 0 < \alpha < p) \qquad \beta \equiv k^{-1}(m - r\alpha) \pmod{\phi(p)}.$$

The message is sent signed as (m, α, β).

The sender of the message can validate the signature by checking if $v_1 \equiv v_2 \pmod{p}$ given the following congruences:

$$v_1 \equiv h^{\alpha}\alpha^{\beta} \pmod{p}, \qquad v_2 \equiv g^m \pmod{p}.$$

If $v_1 \equiv v_2 \pmod{p}$ then, the signature is valid.

5.6 RSA vs ElGamal

People (perhaps unknowingly) have used the RSA cryptosystem if they have "ever used an ATM or purchased something with a credit card over the Internet" [2]. ElGamal is generally used in combination with the Diffie-Hellman key exchange [8]. As stated, the RSA cryptosystem's security relies on the computational infeasibility of factoring the product of two extremely large primes, while the ElGamal cryptosystem relies on the computational infeasibility of solving the discrete

logarithm problem. The RSA cryptosystem discussed in this paper cannot be made any more secure than it already is, as described in this research. That is, the larger p and q, the more secure the cryptosystem is. However, the ElGamal cryptosystem can be made more secure by choosing a random k for every message that is sent and by using discrete logarithms in conjunction with elliptic curves [3]. Note that both the RSA and ElGamal are extremely secure and are virtually equal in security. Their usage in everyday life usually depend on factors that are outside the scope of this research and are more into the topics of computer science.

Speed of computation is something to consider when choosing between the RSA and ElGamal cryptosystems. Multiple studies show that the RSA cryptosystem is faster than ElGamal at encryption/signing and signature verification, but marginally slower at decryption than ElGamal [8][9][10].

6 Conclusion

Cryptography has been in practice for thousands of years, and many different ciphers and cryptosystems have been used throughout history. These ciphers and cryptosystems have also evolved over time, from primitive and insecure methods to those which employ advanced mathematics to secure information. Number theory is one of the more important mathematical fields that has influenced the evolution of cryptography. The early ciphers, like the shift and Vigenère cipher, were created and used without the knowledge that number theory was present in both of their encryption and decryption processes. However, number theory is used extensively in modern day public key cryptosystems like the RSA and ElGamal systems. While these cryptosystems are significantly more secure than their symmetric key predecessors, technology's continual advancements will eventually make these cryptosystems insecure and obsolete. This is why the exploration of the history, evolution, and mathematical concepts behind cryptography is so important. More research needs to be done to further the security and evolution of these cryptosystems in order to protect the welfare of what the cryptosystems are protecting. The reader is encouraged to conduct further research and contribute to the continued development of cryptography for secure information exchange.

7 Diagrams

ASCII Table

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

References

[1] Simon Singh. *The Code Book: The Science of Secrecy from Ancient Egypt to Quantum Cryptography*. Anchor Books, New York, 2000.

[2] Richard E. Klima, Neil P. Sigmon. *Cryptology: Classical and Modern with Maplets*. CRC Press, Boca Raton, 2012.

[3] Wade Trappe, Lawrence C. Washington. *Introduction to Cryptography with Coding Theory*. Second Edition, Prentice Hall, New Jersey, 2006.

[4] David M. Burton. *Elementary Number Theory*. Sixth Edition, McGraw-Hill, New York, 2007.

[5] Whitfield Diffie, Martin Hellman. "New Directions in Cryptography". IEEE Transactions on Information Theory, VOL. IT-22, NO. 6, 644-654, NOV 1976.

[6] T. ElGamal. (1985) "A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms". In: Blakley G.R., Chaum D. (eds) Advances in Cryptology. CRYPTO 1984. Lecture Notes in Computer Science, vol 196. Springer, Berlin, Heidelberg.

[7] R.L. Rivest, A. Shamir, L. Adleman. (1978). "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems". MIT Laboratory for computer Science and Department of Mathematics. Volume 21, number 2.

[8] A.E. Okeyinka. "Computational Speeds Analysis of RSA and ElGamal Algorithms on Text Data". Proceedings of the World Congress on Engineering and Computer Science. Vol I WCECS October 21-23, 2015, San Francisco, USA.

[9] Andysah Putera Utama Siahaan, Elviwani, Boni Oktaviana. "Comparative Analysis of RSA and ElGamal Cryptographic Public-key Algorithms". https://eudl.eu/pdf/10.4108/eai.23-4-2018.2277584

[10] Mahnaz Mohammadi, Alireza Zolghadr, Mohammad Ali Purmina. "Comparison of Two Public Key Cryptosystems". Journal of Optoelectronical Nanostructures, Vol. 3, No. 3. 2018.