1-1-2007

Cryptography in the digital age

Thomas R. Hodge
University of Nevada, Las Vegas

CRYPTOGRAPHY IN THE DIGITAL AGE

by

Thomas R. Hodge, Jr.

Bachelor of Science
Angelo State University, San Angelo, Texas
2002

A thesis submitted in partial fulfillment
of the requirements for the

Master of Science Degree in Mathematical Sciences
Department of Mathematical Sciences
College of Sciences

Graduate College
University of Nevada, Las Vegas
December 2007

Disclaimer

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, the Department of Defense, or the U.S. government.

UNLV
UNIVERSITY OF NEVADA LAS VEGAS

Thesis Approval

The Graduate College
University of Nevada, Las Vegas

November 20, 2007

The Thesis prepared by

Thomas R. Hodge, Jr.

Entitled

Cryptography in the Digital Age

is approved in partial fulfillment of the requirements for the degree of

Master of Science in Mathematical Sciences

Examination Committee Chair

Dean of the Graduate College

Examination Committee Member

Examination Committee Member

Graduate College Faculty Representative

1017-53

ABSTRACT

Cryptography in the Digital Age

by

Thomas R. Hodge, Jr.

Dr. Arthur Baragar, Examination Committee Chair
Professor of Mathematical Sciences
University of Nevada, Las Vegas

Despite it not being an armed conflict between nations, there is a war that has been waged for over 5000 years and is still being fought today. Battles have been won and lost by both sides. The battlefield is the world of cryptography. Our combatants are cryptographers, personnel who make secret codes; and cryptanalysts, personnel who try to break the secret codes.

In this thesis, we examine public-key or asymmetric cryptography, the art of writing or deciphering secret codes or ciphers. We begin by taking a brief look at the overall history of cryptography. Our primary focus involves studying the mathematics behind today's public-key cryptographic methods such as the theory of congruences by Carl Friedrich Gauss, Fermat's little theorem, Euler's phi-function, primitive roots and indices, and elliptic curves over finite fields. Once we have explored the preliminaries we will consider some of the more popular methods of encryption and decryption, for example RSA. We not only discuss how to encrypt and decrypt plain text using these methods but explain why it is hard to break the encrypted text. We conclude our study by inspecting the shortfalls of these ciphers, techniques used to break the encryption, and what the future possibly holds.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

My academic career has not taken the usual path over the last 18 years. I owe two men in the USAF a debt of gratitude. First I owe Colonel Eric Mathewson, who inspired me to leave the comfort of the enlisted ranks and fulfill a lifetime goal. Col. Mathewson always asked for my all and then a little bit more to prove to me I could do anything if I set my mind to it. Col. Mathewson changed the course of my life and I thank him for that. Second I owe Captain William McCulley, my mentor at my first assignment as a commissioned officer. Capt. McCulley urged me to take my education to the next level as soon as possible. I have had several mentors throughout my career, some good, some bad. I consider Capt. McCulley as one of the great ones. What made him so great was he taught me not to stagnate and without exception to give my very best no matter the task.

I would like to thank the Graduate College and the Mathematics Department of UNLV for admitting me as a student. A special thank you to Dr. Zhonghai Ding, who accepted me despite the demanding and short time table. I would also like to thank each member of the committee: Dr. Arthur Baragar, Dr. Peter Shiue, Dr. David Costa, and Dr. Yoohwan Kim.

There are two colleagues who kept my motivation high throughout my stay at UNLV: Major Robert Ain (Ret. USAF) and Steven Fisher. Without their encouragement, the road traveled would have been much harder if not impassable.

Finally I would like to thank my friends and family: Master Sergeant Ron and Patty Hetzel (Ret. USAF), Captain Doug and Jill Mabry (USAF), and my children Amanda, Racheal, Zachary, Jessica, and Alyssa. Last but not least I want to thank my wife, Becca. Whose unconditional support and love helped me to achieve this and much more.

CHAPTER 1

HISTORY OF CRYPTOGRAPHY

Commonly Used Words

Cryptography is a modern word derived from two Greek words "*krypto*" meaning secret or hidden and "*grafo*" meaning to write. However, cryptography is not modern, it started over 4000 years ago with the Egyptians. Before we discuss some of the methods used over the centuries we need a few definitions to make our journey easier.

The first word we need to define is *plaintext.* Plaintext is unencrypted text which is what we are currently reading. The second definition is *ciphertext* which is plaintext that has been encrypted. The process used to encrypt plaintext into ciphertext or vice versa is called a *cipher.* Finally to reverse the process of encryption is called *decryption.*

As we look over cryptography timeline, Tables 1.1-1.3, we focus on a few of the more interesting events of cryptography. For a more in depth view of the history of cryptography a superb choice is *The Codebreakers, The Story of Secret Writing* by David Kahn [11] or with the online *History of Cryptography* by SecureTrust [22]. We begin our venture into history with the Spartans.

Skytale

The Spartans were one of the first civilizations to use cryptography for military purposes. They used a cipher called a skytale. A skytale consists of a wooden dowel or metal rod with a known diameter and a strip of parchment around half inch in width. A Spartan commander

would take the parchment and wrap it around the dowel without overlapping. The message would then be written lengthwise down the dowel, the parchment would be removed, and the ciphertext would be sent to the troops. As seen below in Figure 1.1 the plaintext reads: SEND MORE TROOPS TO SOUTHERN FLANK AND VICTORY IS OURS. The ciphertext reads: STSFVSEROLIONOUACUDOTNTRMPHKOSOSEARRTRNYEONDI.

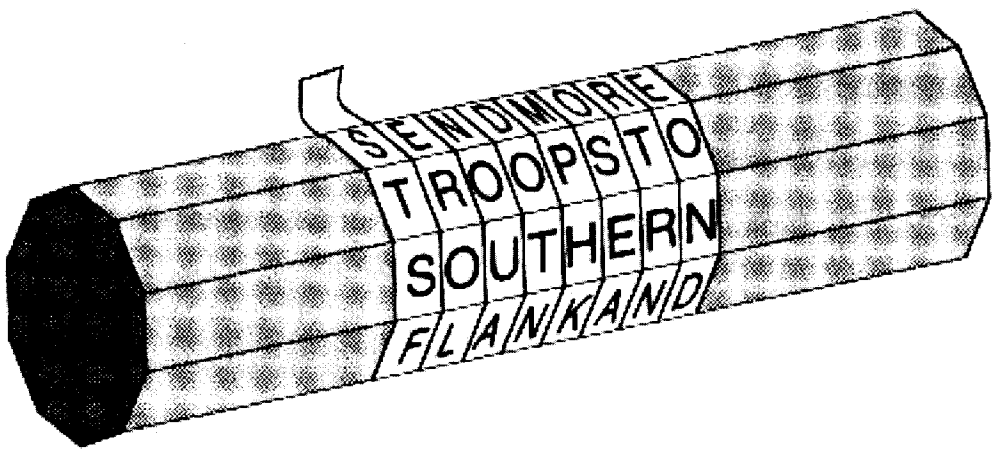

Figure 1.1: Skytale

Caesar's Cipher

The next advance in cryptography happened about 400 years later around 50 BC. Julius Caesar and the Roman Empire were in need of a way to securely communicate with their troops. Caesar developed several cryptographic methods but the most famous is a simple substitution cipher, the Caesar Cipher. The cipher worked by shifting each letter of the Roman alphabet three places as seen in Figure 1.2.

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

Figure 1.2: The Caesar Cipher

If we take our previous example from the Spartans, the plaintext would again read: SEND MORE TROOPS TO SOUTHERN FLANK AND VICTORY IS OURS. The ciphertext using Caesar's cipher would read: VHQG PRUH WURRSV WR VRXWKHUQ IODQN DQG YLFRWUB LV RXUV. Although lacking the complexity of today's modern ciphers, monoalphabetic ciphers like Caesar's cipher remained the norm for hundreds of years. Although there were many advances in the world of cryptography after Caesar's cipher, we are going to fast foward through time to the start of World War II. This is when history saw one of the greatest uses of polyalphabetic ciphers.

World War II

When people think of World War II and cryptography the first thing that usually comes to mind is the German Enigma cipher. However, despite being the last Great War to end all wars, World War II was not a era for new cryptographic methods. Germany's Enigma, Japan's Purple, America's M-209, Britain's TypeX, and other rotor cipher machines were invented between the two great wars. Even the famous Navajo Code-Talkers were initially formed during World War I. So why would we want to look at this era? It produced one invention that has changed the way we live today, the computer.

The ciphertext produced using rotor cipher machines was believed to be unbreakable. A

rotor cipher machine used a similar system to Caesar's cipher, substitution. However, unlike the Caesar cipher, the substitution continually changed thus making it a polyalphabetic cipher. The complexity (number of rotors) and the initial setup had an incomprehensible number of variations leading most to say the ciphertext could not be broken. Young mathematicians from the Polish Cipher Bureau, Marian Rejewski, Henryk Zygalski, and Jerzy Rozycki, were the first to show how to break the Enigma cipher. It was soon discovered that, to decrypt the information in time for it to have value, a faster process must evolve. What they needed was a machine to defeat the machine. Rejewski first invented the cyclometer and later the "bombes" to automate the process of decryption of Enigma traffic. These were the very first types of computers, devices to perform mathematical or logical operations and display the results of those operations.

Breaking the Enigma gave the Allies a tactical advantage but they still lacked a strategic edge. The Axis used the Lorenz rotor machine, not the Enigma, for high-level messages. In 1944, it was the British at Bletchley Park that gave birth to one of the first programmable computers, Colossus. The rough dimension of each Colossus was 50 feet long, 8 feet high, and 3 feet wide. In a matter of hours, Colossus could decrypt messages that would take cryptanalysts weeks to do by hand. Colossus was used to defeat the Lorenz cipher, thus giving the Allies an eye on the inner workings of the Axis leaders, including Hitler. Although computers like Colossus were built in the latter stages of World War II, it would take another invention in 1959 to transfigure computers, the integrated circuit.

The Dawn of a New Age

With the development of computers, like Colossus, the world of cryptography was forever changed. However, it would take two events in recent history to hasten a new age in cryp-

tography. The first was the advent of the integrated circuit in 1959. Before the integrated circuit, computers could easily fill a large room making it improbable for individual ownership. The integrated circuit changed the face of our world in less than twenty years; making computers compact, transportable, and faster. As computers went from just governmental use, to corporations, and finally the mainstream, a need was rising for secure communications. Until 1976, the only way to encrypt information was to use a symmetric-key cipher. Symmetric-key cryptography required both the sender and receiver to know the cipher. If one posses the key (cipher), they would have the ability to encrypt and decrypt the message. This process works fine if there are a limited number of people using the cipher. However, it becomes cost prohibitive to distribute symmetric-keys for encryption to the general public.

In 1976, Dr. Martin Hellman and Whitfield Diffie from Stanford University proposed the second event that forever changed cryptography, a method for public-key cryptography or asymmetric-key cryptography as seen in Figure 1.3. These codes are called asymmetric because having the ability to encrypt a message does not mean one has the ability to decrypt the message. Thus, it was quite simple; two interrelated keys were needed, one public (encryption key) and one private (decryption key). The best part of this system was that there was no need to distribute keys. The public key is used for encryption and generally made available to whomever requires the need for secure communications. The private key is used for decryption and not distributed to the public. One year later, the first of many public-key cryptographic systems was produced. This will be our focus: the mathematics that makes up public-key cryptography, common public-key cryptographic systems, what makes them secure, and finally what the future might hold for cryptography.

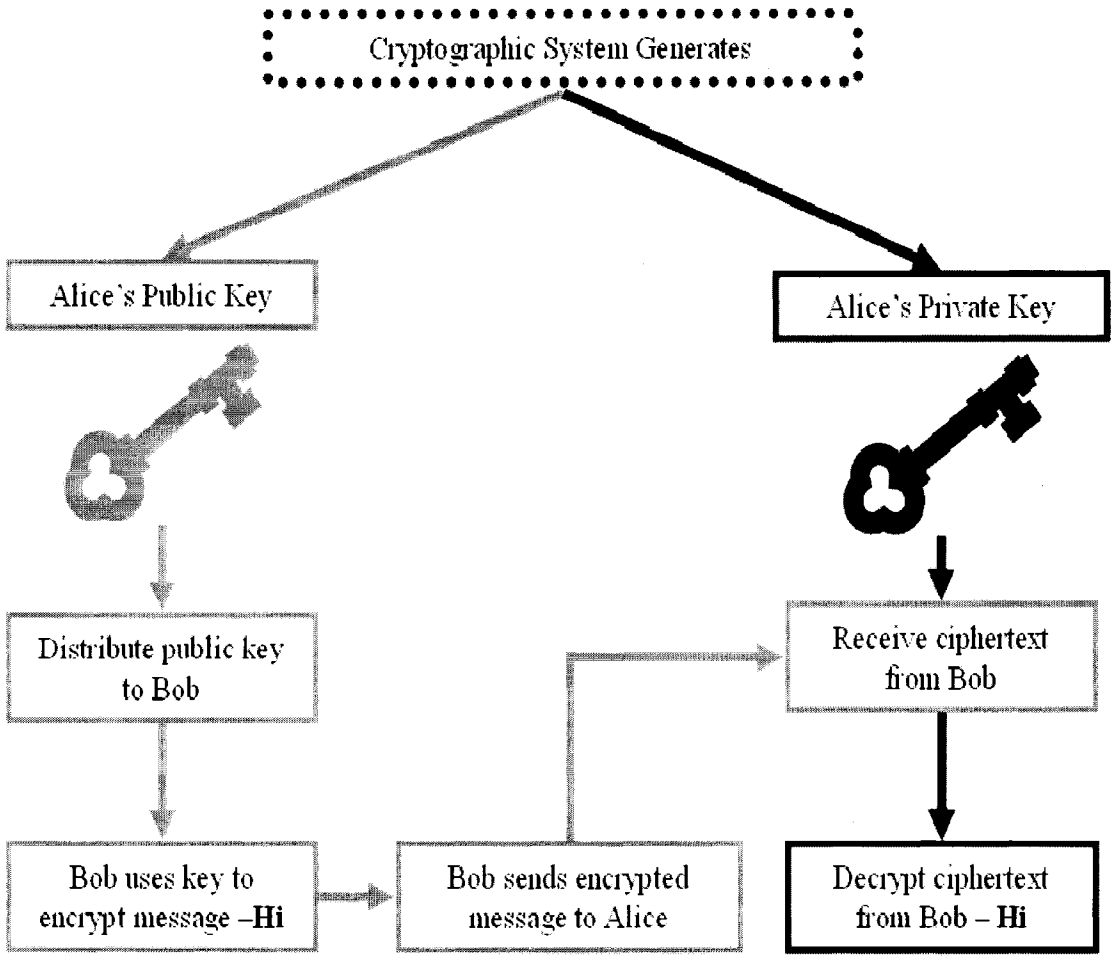

Figure 1.3: Diffie-Hellman's Proposed Scheme

Table 1.1: Cryptography Timeline 1900 BC - 1863

Cryptography Timeline		
Year	*Cipher*	*Remarks*
1900 BC	Hieroglyphics	Hieroglyphic symbols were used by the Egyptians for religious purposes and to record the history of pharaohs. Although the writing was not kept secret, it gave birth to cryptography by transforming words to symbols.
1500 BC	Cuneiform	Mesopotamians used cuneiform writing to protect trade secrets like pottery glazing.
600 BC	Atbash	Hebrew scribes used the atbash cipher to write various parts of the book Jeremiah in the Old Testament. The cipher was a simple substitution cipher: the last letter of the Hebrew alphabet replaced the first and conversely, the second replaced the next to last and conversely, etc.
486 BC	Skytale	The Spartans, military leaders of the Greeks, were the first to develop and employ cryptography solely for military purposes.
50 BC	Caesar	A simple substitution cipher shifting each letter in the alphabet by three places. Any cipher that uses this similar method (shift by four, shift by five, ...) bears the name, Caesar Cipher.
1000	Monoalphabetic	Frequency analysis, determination of how often a letter is used in a specific language, developed as a technique to break monoalphabetic substitution ciphers.
1466	Cipher Disk	Leon Battista Alberti, "The Father of Western Cryptography", developed a cipher disk and cryptographic key using a polyalphabetic substitution.
1587	Vigenere	A polyalphabetic substitution cipher using a short repeating keyword named for Blaise de Vigenere.
1753	Telegraph	Invention of the telegraph was one of the first steps in modern cryptography.
1845	Morse Code	Samuel Morse creates Morse Code, one of the first digital communications.
1863	Vigenere	Prussian Major Kariski proposed a method to break the Vigenere cipher.
Continued on next page...		

Table 1.2: Cryptography Timeline 1883 - 1952

Cryptography Timeline		
Year	*Cipher*	*Remarks*
1883		Auguste Kerckhoff publishes his six principals for military cryptography: 1. The system should be theoretically unbreakable. 2. Compromise of the system should not inconvenience the correspondents. 3. The key should be rememberable without notes and easily changeable. 4. The cryptograms should be transmissible by telegraph. 5. System should be portable and operated by a single person. 6. System should be easy.
1917		The Zimmermann Telegram deciphered by British Intelligence brought the United States into World War I.
1918	ADFGVX	A unique cipher used by the Germans during World War I that used only the letters ADFGVX for all 26 letters of the alphabet.
1918	Enigma	Arthur Scherbius originally designed the Enigma for confidential business communications.
1937	Rotor Machines	Commonly used devices during World War II: Germany - Enigma Japan - Purple US - SIGABA, M-209 Britain - TypeX
1945	Navajo Code Talkers	The Navajo language was used by the US to transmit vital information on Iwo Jima during WW II.
1940's	First Computers	Zuse Z3 - 1941 Atanasoff-Berry - 1941 Colossus - 1944 Harvard Mark I - 1944 ENIAC - 1946
1952	NSA	President Truman created the National Security Agency in 1952.
Continued on next page...		

Table 1.3: Cryptography Timeline 1970 - Present

Cryptography Timeline		
Year	*Cipher*	*Remarks*
1959	Integrated Circuit	Jack Kilby and Robert Noyce developed the integrated circuit revolutionizing the computer industry.
1968	John Walker	The start of John Walker's 17 years of espionage for the Soviets.
1971	Lucifer	Developed by Horst Feistel, it was a simple block cipher that was the predecessor to the Data Encryption Standard (DES). DES was implemented as a standard in 1976.
1974		The National Institute of Standards and Technology was created.
1975	Microsoft	The start up of Microsoft by Bill Gates and Paul Allen.
1976	Diffie-Hellman	Public-key cryptography using a discrete logarithm proposed by Dr. Martin Hellman of Stanford University and Whitfield Diffie.
1977	RSA	Three individuals from MIT, Ron Rivest, Adi Shamir, and Leonard Adleman, conceived of an asymmetric public-key cryptograph system based on factoring a composite number into two exact prime numbers.
1983	Orange Book	DoD STD 5200.28 Trusted Computer System Evaluation Criteria replaced by the ISO 15408 in 2005.
1987	RC4	A stream cipher used to secure wireless networks, internet communication, and point-to-point communication.
1991	PGP	Phil Zimmermann created Pretty-Good-Privacy program to fill the needs of everyday users to encrypt information.
1998	DES	Deep Crack - The Electronic Frontier Foundation cracked the DES algorithm.
2002	AES	Advanced Encryption Standard replaces DES as the new encryption method.
2007	Quantum Computer	The world's first commercial quantum computer demonstrated publicly by D-Wave Systems, Inc. on 13 Feb 2007.

CHAPTER 2

MATHEMATICS

Preliminaries

Since cryptography has become a combination of mathematics and computer science, we need to set the boundaries the two sides can easily work with and understand. The first boundary is the set of integers, {..., -3, -2, -1, 0, 1, 2, 3, ...}; common notation is $\mathbb{Z}$. The second is a subset of the first, the set of natural numbers, {1, 2, 3, ...}; common notation is $\mathbb{N}$. Though 0 is considered by some to be a natural number, we will adopt the convention that it is not in our set. Now that we have set the boundaries, we can state our first theorem, the Division Algorithm:

Theorem 1 (Division Algorithm). *Let $a \in \mathbb{Z}$ and $b \in \mathbb{N}$. Then there exist unique $k, r \in \mathbb{Z}$ satisfying the equation:*

$$a = kb + r \quad where \quad 0 \leq r < b \tag{2.1}$$

Proof. First, let $S = \{a + ib \ : \ a + ib \geq 0, \ i \in \mathbb{Z}\}$. Then by the Well-Ordering Principle there exists a smallest element of S, call it r. We know that $r = a - kb$ for some $k \in \mathbb{Z}$, therefore we have $a = kb + r$.

If we do not have $r < b$, then we have the case $r \geq b$ and we get the following:

$$a - (k+1)b = a - kb - b$$

$$(a - kb) - b = r - b \geq 0.$$

However this implies $a-(k+1)b = r-b < r$, which leads us to a contradiction of the choice of r as the smallest integer, thus $r < b$.

Second, we must show that k and r are unique. Let $a = k_1 b + r_1$ where $0 \leq r_1 < b$ and $a = k_2 b + r_2$ where $0 \leq r_2 < b$. Observe the following two equations:

$$0 \leq r_1 = a - k_1 b, \tag{2.2}$$

$$0 \leq r_2 = a - k_2 b. \tag{2.3}$$

Subtracting equation 2.3 from equation 2.2 we get the following:

$$(r_1 - r_2) = b(k_2 - k_1).$$

Since r_1 and $r_2 < b$ implies $|r_1 - r_2| < b$, we have $b|k_2 - k_1| < b$. We conclude that $k_1 = k_2$ because the only solution for $0 \leq |k_2 - k_1| < 1$ is 0 seeing that $|k_2 - k_1|$ is a non-negative integer. Thus k and r are unique. □

We note that in Equation 2.1, if $r = 0$, then Theorem 1 implies $a = kb$. Equation 2.1 and Theorem 1 gives us two important definitions as seen below.

Definition 1 (Divisor). If $a = kb$ from Equation 2.1 then we say b is a divisor of a or a is a multiple of b, denoted by $b|a$.

Definition 2 (Greatest Common Divisor). Let $a, b \in \mathbb{Z}$, with either $a \neq 0$ or $b \neq 0$. The greatest common divisor of a and b, denoted $\gcd(a, b)$, is a positive integer d that satisfies:

1. $d|a$ and $d|b$.

2. If $c|a$ and $c|b$, then $c \leq d$.

Definition 3 (Least Common Multiple). Let $a, b \in \mathbb{Z}$, with $a \neq 0$ and $b \neq 0$. The least common multiple of a and b, denoted $\text{lcm}\,(a, b)$, is a positive integer m that satisfies:

1. $a|m$ and $b|m$.

2. If $a|c$ and $b|c$ $(c > 0)$, then $m \leq c$.

Theorem 2 (Linear Combination). *Let $a, b \in \mathbb{Z}$, with either $a \neq 0$ or $b \neq 0$. The smallest positive integer of the form $ax + by$, for some $x, y \in \mathbb{Z}$, is $d = \gcd(a, b)$.*

Proof. Let $S = \{ax + by : x, y \in \mathbb{Z}, ax + by > 0\}$ and $d = \min S$.

1. Suppose d is not a divisor of a, then by the division algorithm we know $a = kd + r$ where $0 < r < d$. Solving for r we get the following:

$$r = a - kd = a - k(ax + by) = a(1 - kx) + b(-ky)$$

   This contradicts the fact we choose $d = \min S$, so $d|a$. Similarly, we get $d|b$. Thus d is a common divisor.

2. Suppose $c|a$ and $c|b$. Then $c|ax + by$ for all x, y. In particular $c|d$, therefore $c \leq d$. Thus $d = \gcd(a, b)$. □

This is an existance proof, however it does not provide a way to compute x and y. To do that we use the Euclidean algorithm. First, let us explain how the algorithm works before we prove it. Let $d = \gcd(a, b)$ be our desired outcome. Since $\gcd(|a|, |b|) = \gcd(a, b)$, we can assume without loss of generality $0 < b \leq a$. Applying the Division Algorithm to a and b we get

$$a = k_1 b + r_1 \text{ where } 0 \leq r_1 < b.$$

If $r_1 = 0$ then we can stop and the $\gcd(a,b) = b$. If $r_1 \neq 0$ then we continue the process until a zero remainder appears. This produces a system of equations:

$$\begin{aligned}
a &= k_1 b + r_1 && 0 \leq r_1 < b \\
b &= k_2 r_1 + r_2 && 0 \leq r_2 < r_1 \\
r_1 &= k_3 r_2 + r_3 && 0 \leq r_3 < r_2 \\
&\vdots \\
r_{n-2} &= k_n r_{n-1} + r_n && 0 \leq r_n < r_{n-1} \\
r_{n-1} &= k_{n+1} r_n + 0
\end{aligned} \tag{2.4}$$

We are assured that the sequence has finite number of steps no more than b, because $b > r_1 > r_2 > \cdots \geq 0$ is a decreasing sequence. Thus our claim is the last nonzero remainder, $r_n = d = \gcd(a,b)$. We prove this with the following theorem.

Lemma 3. *If given* $a = kb + r$, *then* $\gcd(a,b) = \gcd(b,r)$.

Proof. Let $d = \gcd(a,b)$ and $\bar{d} = \gcd(b,r)$. We will show $d \leq \bar{d}$ and $\bar{d} \leq d$.

1. $d|\bar{d}$. We are given $d = \gcd(a,b)$. Thus $d|a$ and $d|b$. This implies $d|(a - kb) = r$. Since $\bar{d} = \gcd(b,r)$, we have $d \leq \bar{d}$.

2. $\bar{d}|d$. We are given $\bar{d} = \gcd(b,r)$. Thus $\bar{d}|b$ and $\bar{d}|r$. This imples $\bar{d}|(kb + r) = a$. Since $d = \gcd(a,b)$, we have $\bar{d} \leq d$. □

By applying Lemma 3 repeatedly in the Euclidean algorithm, we get

$$d = \gcd(a,b) = \gcd(b,r_1) = \gcd(r_1,r_2) = \cdots = \gcd(r_{n-1},r_n) = \gcd(r_n,0) = r_n.$$

To get a solution to $ax + by = d$, we work back through the equations in 2.4. This is best

illustrated with an example. Let us now take a look at how the algorithm works by finding $\gcd(16380, 43)$.

$$16380 = 43 \cdot k + r$$

$$16380 = 43 \cdot 380 + 40$$

$$43 = 40 \cdot 1 + 3$$

$$40 = 3 \cdot 13 + 1$$

$$3 = 1 \cdot 3 + 0$$

Therefore the $\gcd(16380, 43) = 1$, so 1 now can be written as a linear combination of 16380 and 43. We start with the equation $40 = 3 \cdot 13 + 1$ and work our way back through system of equations eliminating the remainders.

$$\begin{aligned} 1 &= 40 - (3 \cdot 13) \\ &= 40 - (13 \cdot (43 - (40 \cdot 1))) \\ &= 14 \cdot 40 - 13 \cdot 43 \\ &= 14 \cdot (16380 - 43 \cdot 380) - 13 \cdot 43 \\ &= 14 \cdot 16380 - 5333 \cdot 43 \end{aligned} \tag{2.5}$$

It is obvious that $\gcd(16380, 43) = 1$, because 43 is a prime number. However, the point of the example was to show that using the algorithm gives us a way to write the gcd as a linear combination. Furthermore for our purposes, the algorithm is efficient, requiring at most $2 \log_2 n$ steps [20, page 265]. With the preliminaries out of the way, we are now ready to embark on modular arithmetic introduced by Carl Friedrich Gauss in 1801 [4].

Theory of Congruences

One of the first observations that Gauss made is with Equation 2.1. The equation could be rewritten as $a - r = kb$ and from that we can conclude $b|\,(a - r)$. This led to his definition of congruence as seen in Definition 4:

Definition 4 (Congruence)**.** Let $a, b \in \mathbb{Z}$ and $n \in \mathbb{N}$. The integers a and b are said to be congruent modulo n, denoted by equation

$$a \equiv b \mod n, \tag{2.6}$$

if and only if $n|\,(a - b)$; that is to say $a - b = kn$ for some $k \in \mathbb{Z}$.

Definition 5 (The Set $\mathbb{Z}_n$)**.** Let $\mathbb{Z}_n = \{0, \cdots, n-1\}$. For every $a \in \mathbb{Z}$, $a \equiv b \mod n$ for some $b \in \mathbb{Z}_n$; thus $\mathbb{Z}_n$ is the set of representatives of $\mathbb{Z}$ modulo n.

We also can derive the following properties for Equation 2.6 assuming $a, b, c, d \in \mathbb{Z}$:

1. $a \equiv a \mod n$ (Reflexive).
2. If $a \equiv b \mod n$, then $b \equiv a \mod n$ (Symmetric).
3. If $a \equiv b \mod n$ and $b \equiv c \mod n$, then $a \equiv c \mod n$ (Transitive).
4. If $a \equiv b \mod n$ and $c \equiv d \mod n$, then $(a + c) \equiv (b + d) \mod n$ (Addition).
5. If $a \equiv b \mod n$ and $c \equiv d \mod n$, then $(a - c) \equiv (b - d) \mod n$.
6. If $a \equiv b \mod n$ and $c \equiv d \mod n$, then $ac \equiv bd \mod n$ (Multiplication).
7. If $a \equiv b \mod n$, then $a^k \equiv b^k \mod n$, for any given $k \in \mathbb{N}$.

Proof. We will prove 1, 2, and 4; proofs for the others can be found in most beginning number theory books [4]. (Note: $\Leftrightarrow$ denotes if and only if.)

1. Given any $a \in \mathbb{Z}$, we have $a - a = 0 \cdot n$ thus $a \equiv a \mod n$.

2. Given $a \equiv b \mod n$, we have

$$\begin{aligned} a \equiv b \mod n &\Leftrightarrow n|\,(a-b) \\ &\Leftrightarrow n|\,-(a-b) \\ &\Leftrightarrow n|\,(b-a) \\ &\Leftrightarrow b \equiv a \mod n. \end{aligned}$$

4. Given $a \equiv b \mod n$, we have

$$\begin{aligned} a \equiv b \mod n &\Leftrightarrow n|\,(a-b) \\ &\Leftrightarrow (a-b) = k_1 n \text{ for some } k_1 \in \mathbb{Z}. \end{aligned} \tag{2.7}$$

Also given $c \equiv d \mod n$, we have

$$\begin{aligned} c \equiv d \mod n &\Leftrightarrow n|\,(c-d) \\ &\Leftrightarrow (c-d) = k_2 n \text{ for some } k_2 \in \mathbb{Z}. \end{aligned} \tag{2.8}$$

Adding equations 2.7 and 2.8 we get the following:

$$\begin{aligned} (a-b)+(c-d) = k_1 n + k_2 n &\Leftrightarrow (a+c)-(b+d) = (k_1+k_2)\,n \\ &\Leftrightarrow n|\,(a+c)-(b+d) \\ &\Leftrightarrow (a+c) \equiv (b+d) \mod n. \end{aligned}$$

□

Another observation made by Gauss is his theorem of Linear Congruence.

Theorem 4 (Linear Congruence). *The equation* $ax \equiv b \mod n$ *has a solution if and only if* $\gcd(a, n) \mid b$.

Proof. ($\Rightarrow$) Suppose $ax \equiv b \mod n$. Then $n \mid ax - b$.

$$\Rightarrow ax - b = nk \text{ for some } k \in \mathbb{Z}$$

$$\Rightarrow ax - nk = b.$$

Since $d \mid a$ and $d \mid n$, we get $d \mid b$.

($\Leftarrow$) Suppose $d = \gcd(a, n)$. Then there exists $s, t \in \mathbb{Z}$ such that $as + nt = d$. If $d \mid b$, then there exists $k \in \mathbb{Z}$ such that $b = dk$.

$$\Rightarrow ask + ntk = dk = b$$

$$\Rightarrow x = sk \text{ is a solution.}$$

Thus, we have $ax \equiv b \mod n \Leftrightarrow \gcd(a, n) \mid b$. □

Corollary 5 (Inverse). *If* $\gcd(a, n) = 1$, *then* a^{-1} *exists.*

Proof. By Theorem 4, we can solve $ax \equiv 1 \mod n$. Then the solution is a^{-1}. □

Corollary 6 (Cancellation). *If* $\gcd(c, n) = 1$ *and* $ca \equiv cb \mod n$, *then* $a \equiv b \mod n$.

Proof. We are given $\gcd(c, n) = 1$, so we know c^{-1} exists.

$$ca \equiv cb \mod n \Rightarrow c^{-1}ca \equiv c^{-1}cb \mod n \Rightarrow a \equiv b \mod n.$$ □

Definition 6 (The Set $\mathbb{Z}_n^*$). Let $\mathbb{Z}_n^* = \{a \in \mathbb{Z}_n : \gcd(a, n) = 1\}$. This is the set of $a \in \mathbb{Z}_n$ such that the equation $ax \equiv 1 \mod n$ has a solution.

Gauss made other several significant contributions to the field of number theory, but we are now ready to discuss Fermat's little theorem. To avoid confusion, we adopt Gauss's notation for congruences throughout the rest of this paper.

Fermat's Little Theorem

Although Pierre de Fermat was born over 150 years earlier than Gauss, he too made significant contributions to number theory. Fermat made the following observation about prime numbers as seen in Table 2.1, this led to his little theorem.

Theorem 7 (Fermat's little Theorem). *Let p be a prime number and $a \in \mathbb{Z}_p$ where $a \neq 0$. Then*

$$a^{p-1} \equiv 1 \mod p. \tag{2.9}$$

Corollary 8 (Converse of Theorem 7). *If $a^n \not\equiv a \mod n$ for some a, then n is not prime.*

Note that equation 2.9 can be written as $a^p \equiv a \mod p$ thereby eliminating the requirement of $a \neq 0$. We will not prove Fermat's little Theorem at this moment, we will prove a more general case by Euler later. But as we can see in Table 2.1, the theorem works wonderfully.

Fermat's little Theorem has several applications in the field of number theory. To say the least, it can be used to simplify numbers to a large power congruent modulo some prime. It can be used to solve linear congruences like $ax^t \equiv b \mod p$ or $x^t \equiv b \mod p$. We concluded this section with one important application of Fermat's little Theorem, the so-called "Fermat primality test" of a given number as seen in Corollary 8. One important fact about the test

is that if $a^n \equiv a \mod n$, it does not guarantee n is prime. There exist numbers where the test fails to show they are composite. These numbers are rare (compared to primes) and Fermat's test can be used as an initial first step to determine if a number is prime. However, this is why we should take care in calling it a composite test rather than a primality test.

Table 2.1: Congruence of Prime Numbers

$\mathbb{Z}_3 = \{ 0,1,2 \}$	$\mathbb{Z}_5 = \{ 0,1,2,3,4 \}$	$\mathbb{Z}_7 = \{ 0,1,2,3,4,5,6 \}$	$\cdots$
	$2 \equiv 2 \mod 5$	$2 \equiv 2 \mod 7$	
$2 \equiv 2 \mod 3$	$\vdots$	$\vdots$	$\cdots$
$2^2 \equiv 1 \mod 3 \Leftarrow$	$2^4 \equiv 1 \mod 5 \Leftarrow$	$2^3 \equiv 1 \mod 7 \Leftarrow$	
	$3 \equiv 3 \mod 5$	$3 \equiv 3 \mod 7$	
	$\vdots$	$\vdots$	$\cdots$
	$3^4 \equiv 1 \mod 5 \Leftarrow$	$3^6 \equiv 1 \mod 7 \Leftarrow$	
		$4 \equiv 4 \mod 7$	
	$4 \equiv 4 \mod 5$	$4^2 \equiv 2 \mod 7$	$\cdots$
	$4^2 \equiv 1 \mod 5 \Leftarrow$	$4^3 \equiv 1 \mod 7 \Leftarrow$	
		$5 \equiv 5 \mod 7$	
		$\vdots$	$\cdots$
		$5^6 \equiv 1 \mod 7 \Leftarrow$	
		$6 \equiv 6 \mod 7$	
		$6^2 \equiv 1 \mod 7 \Leftarrow$	$\cdots$

We have covered the basics needed to understand the mathematics used in public-key cryptography. For each cipher we study, we will need to cover more mathematics, but we will postpone covering that background until needed. So, let us look at our first cipher, RSA.

CHAPTER 3

RSA

Euler's Phi-function

Leonhard Euler was one of the greatest mathematicians of the 18th century. He too, like Fermat, made considerable and noteworthy contributions to the field of number theory. Our focus is on Euler's generalization of Fermat's little Theorem. We start by defining Euler's phi-function, how to calculate it, and finally the generalization of Fermat's theorem.

Definition 7 (Euler's Phi-function). We set $\phi(n) = |\mathbb{Z}_n^*|$. That is to say the function denotes the number of positive integers not greater than n that are relatively prime to n.

Note that if n is a prime number p then every positive integer less than p is relatively prime to p. So $\phi(p) = p-1$. Now that we have defined Euler's function we need to know how to calculate it for any $n > 1$. We will need to recall the Fundamental Theorem of Arithmetic, that every positive integer ($n > 1$) can be expressed as a unique product of primes. We are not going to prove them, but the next three important theorems allow us to calculate Euler's function (Proofs can be found in Burton, [4, pages 131-135]).

Theorem 9 (Prime Powers). *If p is a prime and $k > 0$, then $\phi(p^k) = p^k - p^{k-1}$.*

Theorem 10 (Multiplicative). *The phi-function is multiplicative, meaning if $\gcd(m, n) = 1$, then $\phi(mn) = \phi(m)\phi(n)$.*

Proof. Special case: Suppose p and q are distinct primes. Let $n = pq$ and $a \in \mathbb{Z}_n$. If $\gcd(a, n) \neq 1$, then either $p|a$ or $q|a$. There are q values of $a \in \mathbb{Z}_n$ such that $p|a$, namely

$\{0, p, 2p, \cdots, (q-1)p\}$. Similarly, the set of multiples of q is $\{0, q, 2q, \cdots, (p-1)q\}$.

Suppose $kp = lq$ for some $k \in \{0, \cdots, q-1\}$ and $l \in \{0, \cdots, p-1\}$. Then $p|l \Rightarrow l = 0 \Rightarrow k = 0$. So, the overlap is 0. This implies

$$\begin{aligned}\phi(n) = \phi(pq) &= pq - p - q + 1 \\ &= (p-1)(q-1) \\ &= \phi(p)\phi(q).\end{aligned}$$

□

This special case of the multiplicative property of $\phi(n)$ is the one which we most often use.

Corollary 11 (Prime Factorization). *Given* $n = p_1^{k_1} p_2^{k_2} \cdots p_r^{k_r}$ *and* $n > 1$, *then*

$$\phi(n) = \left(p_1^{k_1} - p_1^{k_1-1}\right)\left(p_2^{k_2} - p_2^{k_2-1}\right)\cdots\left(p_r^{k_r} - p_r^{k_r-1}\right). \tag{3.1}$$

We have defined and learned how to solve Euler's phi-function. Our final objective is to state Euler's theorem, a generalization of Fermat's little theorem. Before we state Euler's theorem, we need to recall Corollary 6; if $\gcd(c, n) = 1$ and $ca \equiv cb \mod n$, then $a \equiv b \mod n$.

Theorem 12 (Euler's Theorem). *Let* $n > 1$. *If* $\gcd(a, n) = 1$, *then* $a^{\phi(n)} \equiv 1 \mod n$.

Proof. Let $\{a_1, a_2, \cdots, a_{\phi(n)}\} = \mathbb{Z}_n^*$. Let b_i be the unique integer such that $aa_i \equiv b_i \mod n$, $0 \leq b_i < n$. Since $\gcd(a, n) = \gcd(a_i, n) = 1$, we know $\gcd(b_i, n) = 1$, so $b_i \in \mathbb{Z}_n^*$. Thus $\{b_1, b_2, \cdots, b_{\phi(n)}\} \subseteq \mathbb{Z}_n^*$.

If $b_i \equiv b_j \mod n$ for some $i \neq j$, then $aa_i \equiv aa_j \mod n$. So by cancellation, $a_i \equiv a_j \mod n$ and therefore $a_i = a_j$, a contradiction. Thus $b_i \not\equiv b_j \mod n$, so $\{b_1, b_2, \cdots, b_{\phi(n)}\} = \mathbb{Z}_n^*$. That is $aa_1, aa_2, \cdots, aa_{\phi(n)} \equiv b_1, b_2, \cdots, b_{\phi(n)} \ mod\ n$ in some order. If we multiply these congruences we get the following:

$$\begin{aligned} (aa_1)(aa_2)\cdots\left(aa_{\phi(n)}\right) &\equiv b_1 b_2 \cdots b_{\phi(n)} \mod n \\ a^{\phi(n)}\left(a_1 a_2 \cdots a_{\phi(n)}\right) &\equiv a_1 a_2 \cdots a_{\phi(n)} \mod n \end{aligned}$$

Since $\gcd(a_i, n) = 1$, we can cancel, so $a^{\phi(n)} \equiv 1 \mod n$. □

Note that Euler's theorem implies Fermat's theorem: if n is a prime p, then $a^{\phi(p)} \equiv 1 \mod p \Rightarrow a^{p-1} \equiv 1 \mod p$, which is Equation 2.9.

RSA Cipher

The most well known public-key cryptographic system is RSA. RSA was developed by Ron Rivest, Adi Shamir, and Leonard Adleman one year after Diffe-Hellman's proposal of asymmetric-key cryptography. RSA works because it is easy to multiply two prime numbers together but difficult to factor a number. Let us explore how the cipher works.

Theorem 13 (RSA Algorithm). *We are going to break the algorithm into three parts: the key generation, the encryption process, and the decryption process. The flow of the system can be seen in Figure 1.3 page 6.*

1. *The key generation:*
   (a) *Choose two large random and distinct primes p and q. (Note p and q should be at least 100 digits.)*
   (b) *Compute $n = pq$.*

(c) *Compute* $\phi(n) = (p-1)(q-1)$.

(d) *Choose a random integer* e, *such that* $1 < e < \phi(n)$ *and* $\gcd(e, \phi(n)) = 1$.

(e) *Compute the unique integer* d, *such that* $1 < d < \phi(n)$ *and* $ed \equiv 1 \mod \phi(n)$.

(f) *The public key is* n *and* e; *the private key is* d. *(Note the numbers* p, q, *and* $\phi(n)$ *must be kept secret.)*

2. *Encryption:*

(a) *Bob obtains Alice's public key* n *and* e.

(b) *Bob converts his message to an integer* m, *such that* $0 \leq m \leq n-1$. *If the message is larger than* n, *it can be broken down into pieces such that each piece is within the range. However, each piece* $m_1, m_2, \cdots, m_i$ *will have to be encrypted seperately.*

(c) *Bob computes* $E \equiv m^e \mod n$.

(d) *Bob sends the ciphertext,* E, *to Alice.*

3. *Decryption:*

(a) *Alice receives the ciphertext* E *from Bob.*

(b) *Alice computes* $m \equiv E^d \mod n$ *where* $0 \leq m \leq n-1$ *to decipher the text.*

(c) *Alice converts the integer into text.*

Proof. We are given $\gcd(e, \phi(n)) = 1$ and $ed \equiv 1 \mod \phi(n)$. We claim then $m^{ed} \equiv m \mod n$ for all $m \in \mathbb{Z}_n$. This gives us two cases, $\gcd(m, n) = 1$ and $\gcd(m, n) \neq 1$.

If $\gcd(m, n) = 1$, then Euler's theorem implies $m^{\phi(n)} \equiv 1 \mod n$. We were given $ed \equiv 1 \mod \phi(n) \Leftrightarrow ed = 1 + k\phi(n)$ for some $k \in \mathbb{Z}$. Now $m^{ed} \equiv m^{1+k\phi(n)} \equiv m \cdot m^{k\phi(n)} \equiv m \cdot \left(m^{\phi(n)}\right)^k \equiv m \cdot 1^k \mod n$. Therefore we get our result, $m^{ed} \equiv m \mod n$ when $\gcd(m, n) = 1$.

Although this case is extremely rare, decryption works even if $\gcd(m,n) \neq 1$. If $\gcd(m,n) \neq 1$, then either $p|m$ and the $\gcd(q,m) = 1$ or $q|m$ and $\gcd(p,m) = 1$. Since both these cases are similar and require the same proof we will consider the latter. Now $\gcd(m,n) = q$ implies $q|m$ and $\gcd(p,m) = 1$, in turn this implies $q|m$ and $\gcd(p,m^{q-1}) = 1$. Since p is prime, if we use Euler's theorem we get:

$$\begin{aligned} \left(m^{q-1}\right)^{p-1} &\equiv 1 \mod p \\ m^{(q-1)(p-1)} &\equiv 1 \mod p \\ m^{\phi(n)} &\equiv 1 \mod p \\ \left(m^{\phi(n)}\right)^{k} &\equiv 1^{k} \mod p \end{aligned} \tag{3.2}$$

Multiplying both sides of equation 3.2 by m, we get the result $m^{ed} \equiv m \mod p$ when $q|m$ and $\gcd(p,m) = 1$. By Fermat's theorem we know $\left(m^{ed}\right)^{q} \equiv m^{ed} \mod q$ because q is prime. In this case we have $0^{q} \equiv 0 \mod q$ because $q|m$, so $m^{ed} \equiv m \mod q$. Since $m^{ed} \equiv m \mod p$ and $m^{ed} \equiv m \mod q$, we can conclude by the Chinese remainder theorem (Proof can be found Burton, [4, pages 79-81]) that $m^{ed} \equiv m \mod n$. Similarly if $\gcd(m,n) = p$, we get $m^{ed} \equiv m \mod n$. Therefore $m^{ed} \equiv m \mod n$ for all $m \in \mathbb{Z}_n$. □

We have defined and proved that the RSA cipher works in theory. Let us now take a look at a simple example and show how the cipher works in practice. After choosing our two primes and performing the ground work we will need to convert our message to an integer value. Every letter and symbol on a digital keyboard is already represented by an integer value. However, for our demonstration we will use Table 3.1 to convert our message. Now let us take a look at an example of the RSA cipher.

1. The key generation:

(a) Alice chooses $p = 127$ and $q = 131$. (For this demonstration we will not choose 100 digit primes.)

(b) Alice computes $n = pq = 127 \cdot 131 = 16637$.

(c) Alice computes $\phi(n) = \phi(16637) = (127 - 1)(131 - 1) = 16380$.

(d) Alice chooses $e = 43$.

(e) Alice computes $ed \equiv 1 \mod \phi(n)$, which is $43d \equiv 1 \mod 16380$. Going back to our example in Chapter 2 on the Euclidean algorithm, Equation 2.5, we find that $d = 11047$.

(f) Alice publishes her public key as $n = 16637$ and $e = 43$. Keeping $p = 127, q = 131$, and $\phi(n) = 16380$ secret as well as $d = 11047$, her private key.

2. Encryption:

(a) Bob obtains Alice's public key, $n = 16637$ and $e = 43$.

(b) Bob converts his message "HI" to an integer value using Table 3.1, thus $m = 809$.

(c) Bob computes $E \equiv m^e \mod n$; given $E \equiv 809^{43} \mod 16637$.

$$809^1 \equiv 809 \mod 16637$$

$$809^2 \equiv 5638 \mod 16637$$

$$809^4 \equiv 10374 \mod 16637$$

$$809^8 \equiv 11760 \mod 16637$$

$$809^{16} \equiv 10856 \mod 16637$$

$$809^{32} \equiv 12865 \mod 16637$$

Note that this method of calculating E is referred to as the square-square method.

This gives Bob $809^{43} \equiv 809^{32+8+2+1} \equiv 12865 \cdot 11760 \cdot 5638 \cdot 809 \equiv 1825 \mod 16637$, thus $E = 1825$.

(d) Bob sends $E = 1825$ to Alice.

3. Decryption:

(a) Alice receives $E = 1825$ from Bob.

(b) Alice computes $m \equiv E^d \mod n$ using the square-square method. Alice is given $m \equiv 1825^{11047} \mod 16637$.

$$1825^1 \equiv 1825 \mod 16637$$

$$1825^2 \equiv 3225 \mod 16637$$

$$1825^4 \equiv 2500 \mod 16637$$

$$1825^8 \equiv 11125 \mod 16637$$

$$1825^{16} \equiv 2982 \mod 16637$$

$$1825^{32} \equiv 8166 \mod 16637$$

$$1825^{64} \equiv 2460 \mod 16637$$

$$1825^{128} \equiv 12369 \mod 16637$$

$$1825^{256} \equiv 14946 \mod 16637$$

$$1825^{512} \equiv 14554 \mod 16637$$

$$1825^{1024} \equiv 13269 \mod 16637$$

$$1825^{2048} \equiv 13627 \mod 16637$$

$$1825^{4096} \equiv 9572 \mod 16637$$

$$1825^{8192} \equiv 3225 \mod 16637$$

This gives Alice $1825^{11047} \equiv 1825^{8192+2048+512+256+32+4+2+1} \equiv 3225 \cdot 13627 \cdot 14554 \cdot 14946 \cdot 8166 \cdot 2500 \cdot 3225 \cdot 1825 \equiv 809 \mod 16637$, thus $m = 809$.

(c) Alice uses Table 3.1 to convert the message into text, thus getting 809 = "HI".

Table 3.1: Integer Value Alphabet

A = 01	B = 02	C = 03	D = 04	E = 05	F = 06	G = 07	H = 08
I = 09	J = 10	K = 11	L = 12	M = 13	N = 14	O = 15	P = 16
Q = 17	R = 18	S = 19	T = 20	U = 21	V = 22	W = 23	X = 24
Y = 25	Z = 26						

Security of RSA

We have stated before that the security of RSA lies in the fact that it is easy to calculate a composite number with two distinct primes but hard to factor it. Despite not having an efficient method to factor a composite number, we cannot say RSA is invulnerable to attack. We turn our attention to the various methods used to factor n.

Before we start examining the time it takes various methods used to factor n, let us recall the greatest common divisor. If we are given two positive integers, m and n such that $n \leq m$, then the time it takes to compute the $\gcd(n, m)$ using the simplest method of divide and check is at most $2n$. Using the common notation of "big O" (See Koblitz, [12, page 9]), then we can say it takes $O(n)$. Yet in Chapter 2 we discovered a more efficient method to calculating the gcd, it is the Euclidean Algorithm. The time it takes for the Euclidean Algorithm is $O(\log n)$. If we are given a positive integer n, the simplest way to check if it is prime is again the check and divide method. Since we only have to check

primes up to $\sqrt{n}$, we do $O(\sqrt{n})$ calculations [19]. Modifying this method using only prime numbers to divide and check against $n = pq$, we can break RSA. This seems quite feasible when we look back on our example; it would take less than 128 steps to complete. If we started at 2 and tested using only prime numbers up to 127, we would find that it only took 31 steps to break our RSA demonstration. Still, we took the liberty to modify the first step to show a working example. If we were to choose p and q such that they are at least 100 digit prime numbers, then our simple method of trying to break RSA would be computationally unrealistic. Various methods have been used to factor numbers less than 100 digits such as quadratic sieve, Fermat factorization, continued fractions, elliptic curves, and the rho method. But, the best time that any of these methods could produce to factor n is $\exp\left(O\left(\sqrt{\log n \log\log n}\right)\right)$ [12].

On March 18, 1991 RSA Laboratories developed the RSA Factoring Challenge to monitor the progress in the development of integer factorization methods. RSA-100 through RSA-155, integers ranging from 100 to 155 digits composed of two large primes, were cracked using either the quadratic sieve or the number field sieve before the year 2000. (Note that RSA-155, 155 digits, is a 512-bit key.) During this time frame a new method of factorization was developed, the general number field sieve, and it could handle numbers larger than 155 digits. The speed of the general number field sieve is $\exp\left(O\left((\log n)^{1/3}(\log\log n)^{2/3}\right)\right)$ [12]. Even though this method is faster than the others and far better than the brute force method of guessing the private-key, it still requires a large amount of computing power and time for very large integers. With these advancements in factoring, RSA ended the challenge in 2007. Currently it is recommended to use key lengths of 1024-bits for general security and 2048-bits for more valuable information [19].

Cryptanalysis

What do we do if we are not just dealing with RSA encryption? We must use some form of cryptanalysis to break the cipher. The first requirement of cryptanalysis is to have a working expertise of the cipher being used. The second requirement is the gathering of the plaintext, the ciphertext, or both. This method of trying to break the cipher depends on time, patience, knowledge, and sometimes luck. Let us take a look at the four main methods used by cryptanalysts:

1. Ciphertext-only attack. This type of attack calls for the cryptanalyst to have access to several messages that have been encrypted using the same cipher. The job is to decipher as many of the messages as possible and ultimately attain the secret keys.

2. Known-plaintext attack. This type of attack the cryptanalyst has access to both the ciphertext and the plaintext. The job is to retrieve the secret key.

3. Chosen-plaintext attack. This type of attack is very similar to the known-plaintext attack, but it carries a bigger punch. The reason this attack is more powerful is because a cryptanalyst can now choose a specific plaintext to be encrypted. Again the supreme goal here is to procure the secret key.

4. Adaptive-chosen-plaintext attack. As the name entails, this attack is similar to the last, yet even more powerful. Unlike the last, the cryptanalyst now has the ability to vary the length of the plaintext to be encrypted as well as choose a second plaintext to be encrypted based on the first.

Although these types of attacks on RSA do not work (they are essentially equivalent to factoring n), they are still useful on other ciphers. We only brushed the surface of cryptanalysis. A cryptanalysts carries a variety of other tools in their toolbox including: chosen-ciphertext

attack, chosen-key attack, rubber-hose cryptanalysis, differential cryptanalysis, and linear cryptanalysis. For more details see *Applied Cryptography* by Bruce Schnier [21].

As keys get longer, another option to break the encryption is a side-channel attack. This type of attack does not try to solve the inverse of a cipher but rather goes after the implementation of the cryptographic system. Rather than just focusing on RSA, side-channel attacks can be used to attempt to break most of today's ciphers. So, what is a side-channel attack? The focus of side-channel attack is the time it takes to perform the calculations, power used during calculations, sound made performing calculations, or electro-magnetic radiation. Although this attack is dependent on a lot of variables such as CPU used, type of countermeasures used including shielding, accuracy of measuring equipment, and various others, it still is a feasible method to break a cipher. Side-channel attacks allow cryptanalysts to discern information about the secret-key by careful measurement of time, power, sound, or electro-magnetic radiation; thereby narrowing their focus on deducing the secret-key. Even though this method requires less computing power and time, it still requires analysts to interpret the measurements to comprehend relevant information about the cipher. Ways to reduce these attacks include adding delays, power balancing, adding noise, and shielding; but in turn this raises the cost, time, or size of the equipment needed to perform the cipher.

In today's world, we have become reliant on computers and the internet for entertainment, communication, and business. We trust corporations to protect our private information using some type of security such as symmetric-key or asymmetric-key cryptography. Indeed if we were unable to break the cipher using the above two methods, we could consider an even more simplistic attack bypassing the security of the cipher, an attack on the computer system. In partnership with the Department of Homeland Security, the United States Computer Emergency Readiness Team (US-CERT) is charged with defending our cyberspace. In 2006,

US-CERT listed over 450 vulnerabilities that could be exploited to gain unauthorized access to a computer system [24]. The majority of these vulnerabilities can be eliminated or the risk mitigated. Cyber criminals are not just putting viruses on our computer systems; they are using sophisticated tools for phishing, spoofing, and pharming to try to gain access to our personal information. Although this risk is low for individuals, it has been steadily increasing over the last few years. We are only covering the surface of this risk because it could lead to the compromise of the secret key. Modifying an old proverb we can say that cryptography is only as strong as its weakest link. In today's world the weakest link is no longer the cipher, but rather the protection of the secret key.

CHAPTER 4

EL GAMAL (ELG)

Primitive Roots and Indices

Although primitive roots and indices can be defined for any positive integer (though they do not always exist) we limit ourselves to prime numbers when using the ElG cipher. We start by defining what a primitive root is. Next, we establish how many primitive roots there are for each prime. Finally, we define index and state three properties used in arithmetic.

Definition 8 (Order). Let $a \in \mathbb{Z}_n^*$. The order of an integer a modulo n is the smallest positive integer, k, such that $a^k \equiv 1 \mod n$; denoted as $\text{ord}(a) = k$.

Definition 9 (Primitive Root). Let $\mathbb{Z}_p^* = \{1, 2, 3, \cdots, p-1\}$, where p is prime. If there exists a $g \in \mathbb{Z}_p^*$ such that $\{g^1, g^2, g^3, \cdots, g^{p-1}\} = \mathbb{Z}_p^*$ (i.e. $\text{ord}(g) = \phi(p)$), then g is called a primitive root or a generator of $\mathbb{Z}_p^*$.

We soon will discover that $\mathbb{Z}_p^*$ always has a primitive root.

Lemma 14. *Suppose we are given a polynomial $f(x)$ that has n distinct roots in $\mathbb{Z}_p$. Then* $\deg(f) \geq n$.

Proof. We prove this by induction on n.

- Initial case: Suppose $f(x)$ has one root. This clearly implies $\deg(f) \geq 1$.
- Induction: Let us assume the Lemma is true when $n = k - 1$. Suppose $f(x)$ has k distinct roots $\{r_1, \cdots, r_k\}$. We need to show $\deg(f) \geq k$. Since $f(r_k) \equiv 0 \mod p$, we

know $(x - r_k) | f(x)$. This implies $f(x) \equiv (x - r_k) g(x) \mod p$ for some $g(x)$. Since $f(r_i) \equiv 0 \mod p$ for $i = \{1, \cdots, k-1\}$, we know $(r_i - r_k) g(r_i) \equiv 0 \mod p$. This implies $p | (r_i - r_k) g(r_i)$. Because $r_i - r_k \not\equiv 0 \mod p$, since the roots are distinct, we get $p | g(r_i)$. That is to say $g(x)$ has roots $\{r_1, \cdots, r_{k-1}\}$. So by the induction hypothesis, $\deg(g) \geq k - 1$; therefore $\deg(f) \geq k$. □

Lemma 15. *Suppose* $ord(a) = r$, $ord(b) = s$, *and* $\gcd(r, s) = 1$. *Then* $ord(ab) = rs$.

Proof. Since $a^r \equiv 1 \mod p$ and $b^s \equiv 1 \mod p$, we get $(ab)^{rs} \equiv (a^r)^s (b^s)^r \equiv 1 \mod p$. Suppose

$$(ab)^k \equiv 1 \mod p.$$

Then

$$(ab)^{kr} \equiv b^{kr} \equiv 1 \mod p$$

Since $\gcd(r, s) = 1$, this implies $s | k$. Similarly, we get $r | k$. So, $rs | k$; therefore $rs \leq k$. □

Lemma 16. *Let* $r = max\{ord(a) : a \in \mathbb{Z}_p^*\}$. *If* $b \in \mathbb{Z}_p^*$, *then* $ord(b) | r$.

Proof. Suppose there exists b such that $\text{ord}(b) \nmid r$. Then there exists a prime $p | \text{ord}(b)$ such that $p^m | \text{ord}(b)$ but $p^m \nmid r$. Let n be the maximum integer such that $p^n | r$ (so $n < m$) and let $r = p^n \hat{r}$. We were given there exists an $a \in \mathbb{Z}_p^*$ such that $\text{ord}(a) = r$. Let $s = \frac{\text{ord}(b)}{p^m}$, $\hat{b} = b^s$, and $\hat{a} = a^{p^n}$. Then $\text{ord}(\hat{a}) = \hat{r}$, $\text{ord}\left(\hat{b}\right) = p^m$, and the $\gcd(p^m, \hat{r}) = 1$. So by Lemma 15, $\text{ord}\left(\hat{a}\hat{b}\right) = p^m \hat{r} > r$, a contradiction. □

Theorem 17 (Primitive Root). *Every* $\mathbb{Z}_p^*$ *has a primitive root.*

Proof. Let $k = \max\{\text{ord}(a) : a \in \mathbb{Z}_p^*\}$. Then by Lemma 16, $a^k \equiv 1 \mod p$ for all a. This implies $f(x) = x^k - 1$ has $p - 1$ roots. In turn, Lemma 14 implies $k = \deg(f) \geq p - 1$. Since $a^{p-1} \equiv 1 \mod p$ for all a, $k = p - 1$. Therefore a primitive root exists for $\mathbb{Z}_p^*$. □

We have established that a primitive root exists for $\mathbb{Z}_p^*$, but how many are there? Our next theorem will determine the number of primitive roots.

Theorem 18 (Number of Primitive Roots). *There are exactly $\phi(p-1)$ primitive roots of $\mathbb{Z}_p^*$.*

Proof. Let r be an integer and suppose $\gcd(r, p-1) = d$ and $d > 1$. Let $p-1 = dk$ and $r = \hat{r}d$. Then $(g^r)^k \equiv g^{\hat{r}dk} \equiv g^{\hat{r}(p-1)} \equiv 1 \mod p$. This implies $\text{ord}(g^r) < p-1$. Thus there are at most $\phi(p-1)$ primitive roots.

Conversely, if $\gcd(r, p-1) = 1$, then $rx + (p-1)y = 1$. Thus $(g^r)^k \equiv 1 \mod p$ implies $g^{rkx+(p-1)ky} \equiv g^k \equiv 1 \mod p$. In turn, this implies $p-1|k = \text{ord}(g)$. Thus g^r is a primitive root. □

We state our next theorem to test for a primitive root rather than using the naive approach of calculating $a^k \mod p$ for every $k = \{1, \cdots, p-1\}$.

Theorem 19 (Primitive Root Test). *Let $a \in \mathbb{Z}_p^*$ and q prime. If $a^{\frac{p-1}{q}} \not\equiv 1 \mod p$ for all $q|p-1$. Then a is a generator.*

Proof. Suppose $a^{\frac{p-1}{q}} \not\equiv 1 \mod p$ for all $q|p-1$. Let $\text{ord}(a) = k$, and suppose $k < p-1$. By Lemma 16 $k|p-1$. So $p-1 = kt$ for some $t \in \mathbb{Z}$. Since $k \neq p-1$, $t > 1$, so there is a prime $q|t$. But then $t = qt'$ and $a^{kt'} \equiv 1 \mod p$. This implies $a^{\frac{p-1}{q}} \equiv 1 \mod p$, a contradiction. □

Though primitive roots are plentiful, there is no simple method to decide whether a number is a primitive root for a large prime. (To apply Theorem 4.1, we need to factor $p-1$, which is difficult if $p-1$ has a large prime factor [12].) Once one is found though, the rest can easily be calculated using the formula $g^s \mod p$ such that $\gcd(s, \phi(p)) = 1$. Let us look at an example; let us find the primitive roots of $\mathbb{Z}_{17}^*$.

1. First, we find a primitive root.

(a) First we start with $g = 2$ and determine if it is a generator.

$$2^{\frac{17-1}{2}} \equiv 2^8 \equiv 1 \mod 17.$$

(b) Since 2 is not a generator, we must test again. Let us try $g = 3$.

$$3^{\frac{17-1}{2}} \equiv 3^8 \equiv 16 \not\equiv 1 \mod 17.$$

So by Theorem 4.1, 3 is a primitive root.

2. We have established a generator, $g = 3$; so, how many are there?

$$\phi(17-1) = \phi(16) = 2^4 - 2^3 = 16 - 8 = 8$$

3. What are the generators?

(a) Since $g = 3$, the number of s relatively prime to $\phi(17)$ is 8, and they are $s = \{1, 3, 5, 7, 9, 11, 13, 15\}$.

(b) Using our formula, we calculate all the primitive roots:

$$3^1 \equiv 3 \mod 17$$

$$3^3 \equiv 10 \mod 17$$

$$3^5 \equiv 5 \mod 17$$

$$3^7 \equiv 11 \mod 17$$

$$3^9 \equiv 14 \mod 17$$

$$3^{11} \equiv 7 \mod 17$$

$$3^{13} \equiv 12 \mod 17$$

$$3^{15} \equiv 6 \mod 17$$

Thus the set of primitive roots for $\mathbb{Z}_{17}^*$ is $\{3, 5, 6, 7, 10, 11, 12, 14\}$.

As Table 4.1 suggests, it is best to start testing generators at 2 and work our way up. Regardless, we should not be fooled into believing that all least primitive roots are small; for example if $p = 191$, the smallest generator is $g = 19$ [4].

We started this section with primitive roots so that we could analyze index calculus, leading us to our definition of index.

Definition 10 (Index). If r is a primitive root of p and $r^k \equiv a \mod p$, then k is called the *index of a relative to r*, denoted by $k = \text{ind}_r a$; hence

$$r^{\text{ind}_r a} \equiv a \mod p. \tag{4.1}$$

The properties of indices are very similar to the properties of logarithms. The properties of indices are as follows, given r is a primitive root of p:

1. $\text{ind}_r ab \equiv \text{ind}_r a + \text{ind}_r b \mod \phi(p)$
2. $\text{ind}_r a^t \equiv t \, \text{ind}_r a \mod \phi(p)$
3. $\text{ind}_r 1 \equiv 0 \mod \phi(p)$ and $\text{ind}_r r \equiv 1 \mod \phi(p)$

Table 4.1: Primitive Roots for $1 < p < 100$

Prime	Number of Generators	Smallest Primitive Root
2	1	1
3	1	2
5	2	2
7	2	3
11	4	2
13	4	2
17	8	3
19	6	2
23	10	5
29	12	2
31	8	3
37	12	2
41	16	6
43	12	3
47	22	5
53	24	2
59	28	2
61	16	2
67	20	2
71	24	7
73	24	5
79	24	3
83	40	2
89	40	3
97	32	5

Proof. 1. By the definition of index we have:

$$r^{\text{ind}_r a} \equiv a \mod p$$

$$r^{\text{ind}_r b} \equiv b \mod p$$

If we multiply these two equations together, we get the following:

$$r^{\,\mathrm{ind}_r a}\, r^{\,\mathrm{ind}_r b} \equiv ab \mod p$$

$$r^{\,\mathrm{ind}_r a + \mathrm{ind}_r b} \equiv r^{\,\mathrm{ind}_r ab} \mod p$$

$$r^{\,\mathrm{ind}_r a + \mathrm{ind}_r b - \mathrm{ind}_r ab} \equiv 1 \mod p$$

Recall that the order of r is $\phi(p)$, therefore

$$\phi(p) \mid \mathrm{ind}_r a + \mathrm{ind}_r b - \mathrm{ind}_r ab \Leftrightarrow \mathrm{ind}_r a + \mathrm{ind}_r b - \mathrm{ind}_r ab \equiv 0 \mod \phi(p)$$

$$\Leftrightarrow \mathrm{ind}_r a + \mathrm{ind}_r b \equiv \mathrm{ind}_r ab \mod \phi(p).$$

2. By the definition of index we have:

$$r^{\,\mathrm{ind}_r a^t} \equiv a^t \mod p$$

By the laws of exponents we have $r^{t\,\mathrm{ind}_r a} \equiv \left(r^{\,\mathrm{ind}_r a}\right)^t \equiv a^t \mod p$, therefore again by the order of r we have:

$$\mathrm{ind}_r a^t \equiv t\,\mathrm{ind}_r a \mod \phi(p).$$

3. Let $k = \mathrm{ind}_r 1$, so by definition we have $r^k \equiv 1 \mod p$. Since k is a multiple of $\phi(p)$, we get $k \equiv 0 \mod \phi(p)$.

   Let $k = \mathrm{ind}_r r$, so by definition we have $r^k \equiv r \mod p$ or $r^{k-1} \equiv 1 \mod p$. Since $k-1$ is a multiple of $\phi(p)$, we get $k \equiv 1 \mod \phi(p)$. □

ElG Cipher

Like the RSA cipher, the ElG is a public-key cryptosystem that uses large primes as a cipher, key exchange, or digital signature. However RSA's security is in the quandary of factoring, where as ElG's security lies in the difficulty of the discrete log problem. That is to say, the difficulty of solving for s in $x^s \equiv y \mod p$ given x, y and p (i.e. finding $s = \text{ind}_x y$). Let us analyze how the ElG cipher works.

Theorem 20 (ElG Algorithm). *Again, we break the algorithm into three parts: the key generation, the encryption process, and the decryption process.*

1. *The key generation:*
   (a) *Choose a large prime, p (at least 100 digits).*
   (b) *Choose an $\alpha \in \mathbb{Z}_p^*$ (preferably with α a primitive root[4]).*
   (c) *Choose a random integer, a, such that $1 < a < p-1$.*
   (d) *Compute $\beta \equiv \alpha^a \mod p$.*
   (e) *The public key is p, α, and β. The private key is a.*
2. *Encryption:*
   (a) *Bob obtains Alice's public key p, α, and β.*
   (b) *Bob converts his message to an integer m, such that $0 \leq m \leq p-1$. If the message is larger than p, it can be broken down into pieces such that each piece is within the range. However, each piece $m_1, m_2, \cdots, m_i$ will have to be encrypted separately.*
   (c) *Choose a random integer, k, such that $1 < k < p-1$. If the message is broken down into pieces $m_1, m_2, \cdots, m_i$, then a different random k should be chosen for each m_i [23].*

(d) Bob computes $\gamma \equiv \alpha^k \mod p$.

(e) Bob computes $\delta \equiv m\beta^k \mod p$.

(f) Bob sends the ciphertext, $E = (\gamma, \delta)$*, to Alice.*

3. Decryption:

(a) Alice receives the ciphertext E *from Bob.*

(b) Alice computes $\chi \equiv \gamma^{p-1-a} \equiv \gamma^{-a} \mod p$

(c) Alice computes $m \equiv \delta\chi \mod p$ *to decipher the text.*

(d) Alice converts the integer into text.

Proof. $\delta\chi \equiv \delta\gamma^{-a} \equiv \delta\alpha^{-ak} \equiv m\beta^k\alpha^{-ak} \equiv m\left(\alpha^{ak}\right)\left(\alpha^{-ak}\right) \equiv m \mod p$ □

We have defined and proved that the ElG cipher works in theory. Let us demonstrate the procedure with an example (but with a small prime).

1. The key generation:

   (a) Alice chooses a small prime, $p = 821$. (For this demonstration we will not choose a large prime.)

   (b) Alice chooses a primitive root, $\alpha = 2$.

   (c) Alice chooses a random integer, $a = 10$, such that $1 < a < 820$.

   (d) Alice computes $\beta \equiv \alpha^a \mod p$, giving $\beta \equiv 2^{10} \mod 821$.

$$2^1 \equiv 2 \mod 821$$

$$2^2 \equiv 4 \mod 821$$

$$2^4 \equiv 16 \mod 821$$

$$2^8 \equiv 256 \mod 821$$

This gives Alice $2^{10} \equiv 2^{8+2} \equiv 256 \cdot 4 \equiv 1024 \equiv 203 \mod 821$. Thus $\beta = 203$.

(e) The public key is $p = 821, \alpha = 2$, and $\beta = 203$. The private key is $a = 10$.

2. Encryption:

(a) Bob obtains Alice's public key $p = 821, \alpha = 2$, and $\beta = 203$.

(b) Bob converts his message "Hi" to an integer value using Table 4.2, thus $m = 809$.

(c) Bob chooses a random integer, $k = 100$, such that $1 < k < 820$.

(d) Bob computes $\gamma \equiv \alpha^k \mod p$, giving $\gamma \equiv 2^{100} \mod 821$. Using the square-square method Bob concludes $\gamma \equiv 2^{100} \equiv 784 \mod 821$, thus $\gamma = 784$.

(e) Bob computes $\delta \equiv m\beta^k \mod p$, giving $\delta \equiv 809 \cdot 203^{100} \mod 821$. Again using the square-square method Bob concludes $203^{100} \equiv 463 \mod 821$. Thus $\delta \equiv 809 \cdot 203^{100} \equiv 809 \cdot 463 \equiv 191 \mod 821$, so $\delta = 191$.

(f) Bob sends the ciphertext, $E = (\gamma = 784, \delta = 191)$, to Alice.

3. Decryption:

(a) Alice receives the ciphertext E from Bob.

(b) Alice computes $\chi \equiv \gamma^{p-1-a} \equiv \gamma^{-a} \mod p$, giving $\chi \equiv 784^{821-1-10} \mod 821$. Using the square-square method Alice concludes $\chi \equiv 784^{810} \equiv 649 \mod 821$, thus $\chi = 649$.

(c) Alice computes $m \equiv \delta\chi \mod p$ to decipher the text giving $m \equiv 191 \cdot 649 \mod 821$. Thus $m = 809$.

(d) Alice uses Table 4.2 to convert the message into text, thus getting 809 = "HI".

Table 4.2: Integer Value Alphabet

A = 01	B = 02	C = 03	D = 04	E = 05	F = 06	G = 07	H = 08
I = 09	J = 10	K = 11	L = 12	M = 13	N = 14	O = 15	P = 16
Q = 17	R = 18	S = 19	T = 20	U = 21	V = 22	W = 23	X = 24
Y = 25	Z = 26						

Security of ElG

Like the RSA cipher, it has been conjectured that the ElG cipher is safe to use for public-key cryptography, if used properly [12]. ElG's security lies in the discrete log problem, the difficulty of solving for a in $\alpha^a \equiv \beta \mod p$. If we were to try and break our ElG cipher by using index calculus, the naive approach would be to build an index table, which in our example above, requires 820 calculations. Let us look at a smaller example to illustrate how it could be done.

1. Suppose we discover that an ElG public key is $(p = 17, \alpha = 7, \beta = 10)$ and we want to break it.

2. Breaking Our Cipher:

   (a) We first find a generator of $\mathbb{Z}_{17}^*$. From our previous work, we know $g = 3$.

   (b) We calculate $3^k \mod 17$, $k = \{1, \cdots, p-1\}$, to generate the following index table.

Table 4.3: Index Table for $p = 17$ and $g = 3$

a	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$\text{ind}_3 a$	16	14	1	12	5	15	11	10	2	3	7	13	4	9	6	8

(c) We know $p = 17, \alpha = 7$, and $\beta = 10$. Thus $7^a \equiv 10 \mod 17$. So we need to find a:

$$\begin{aligned} \text{ind}_3 7^a &\equiv \text{ind}_3 10 \mod 16 \\ a\ \text{ind}_3 7 &\equiv \text{ind}_3 10 \mod 16 \\ 11a &\equiv 3 \mod 16 \end{aligned}$$

(d) We now must solve $11a - 16b = 3$ using the Euclidean algorithm.

$$\begin{aligned} 16 &= 11 \cdot 1 + 5 \\ 11 &= 5 \cdot 2 + 1 \\ 5 &= 1 \cdot 5 + 0 \end{aligned}$$

Working our way back we get $1 = 11 \cdot 3 - 16 \cdot 2$. If we multiply by 3, we get $3 = 11 \cdot 9 - 16 \cdot 6$.

(e) Therefore, the secret key is $a = 9$.

To break the cipher above, we generated an index table. Consequently if we choose a large p, say at least 100 digits, it now becomes infeasible to try to break the ElG cipher using this method. As with RSA, algorithms have been developed to solve the discrete log problem for a large p (e.g. Shanks's baby-step giant-step, Pollard's ρ-method, Pohlig-Hellman method). The fastest of these methods, Pohlig-Hellman, still takes $O\left(\sqrt{p}\log p\right)$ to solve the discrete log problem [8]. Like RSA, the ElG cipher is resistant to cryptanalysis, but vulnerable to side-channel attacks and cyber criminals.

CHAPTER 5

ELLIPTIC CURVES

Both the RSA cipher and the ElG cipher have been in use for over twenty years and are considered first generation public-key cryptosystems. Both of these systems require large secret keys (i.e. greater than 100 digits) to maintain the security of the information. Currently it is recommended to use a key size of 1024-bits, over 300 digits, for RSA encryption. The next generation of public-key cryptography is elliptic curves over finite fields. The security of elliptic curve cryptography (ECC) lies in the elliptic curve discrete log problem. It is easy for us to solve for Q in $[k]\,P = Q$ given k and P, but extremely difficult to calculate k given P and Q. Like ElG, we use this as our basis. Thus, we make the elliptic curve E defined over $\mathbb{F}_q$ and a point P on E public; Alice picks u and sends Bob $[u]\,P$, Bob picks v and sends Alice $[v]\,P$. Then, as with ElG, Alice and Bob can calculate $[uv]\,P$, their shared key. The biggest advantage of using ECC is key size. RSA requires a key of 1024-bits for security, whereas for ECC to provide the same security only requires a key size of 160-bits, less than 50 digits. ECC is done over the finite field, $\mathbb{F}_q$ where $q = p^r$ is large. Though the choice of $q = 2^r$ is common, for simplicity we will restrict our attention to $q = p$ with p a large prime (50-100 digits).

Elliptic Curves over the Reals

To gain a working knowledge of ECC over $\mathbb{F}_q$, we first must understand how elliptic curves perform over the reals. Elliptic curves are cubic curves of degree three in two variables chosen

so that a line that intersects the curve at two points intersects the curve again at a third (counting multiplicity). By a suitable change of variables, every elliptic curve over the reals is equivalent to one in the form:

$$y^2 = x^3 + ax + b \quad \text{for some } a, b \in \mathbb{R}. \tag{5.1}$$

This is called the Weierstrass canonical form. We limit ourselves to looking at only smooth curves; that is, curves where a tangent line can be found for any point on it. This reduces to the condition $4a^3 + 27b^2 \neq 0$. A generic example is shown in Figure 5.1. Note that a line going through any two points on the curve generally intersects the curve at a third point; the exception is if the line is vertical. To deal with the case of vertical lines, we add a point at infinity.

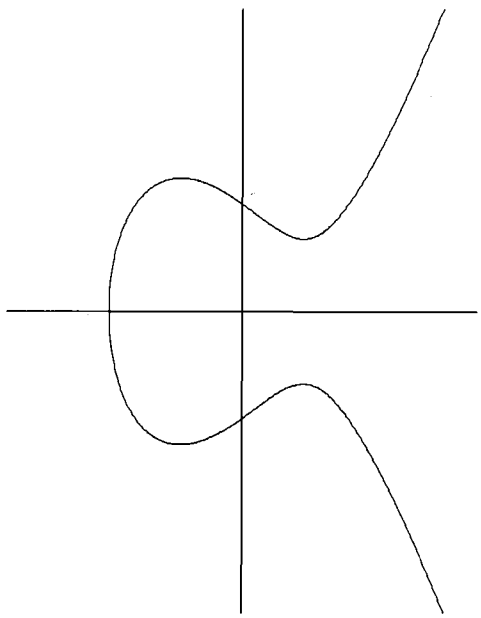

Figure 5.1: An Elliptic Curve

Definition 11 (Point at Infinity). Let $\mathcal{O}$ be the point at infinity that all vertical lines go through.

This definition can be made more precise by using projective geometry (See Koblitz [12, page 171]). We state the following property of $\mathcal{O}$ without proof: $\mathcal{O}$ is a point of inflection, meaning the tangent line at $\mathcal{O}$ intersects the curve again at $\mathcal{O}$. We call all the points that satisfy Equation 5.1 plus the point $\mathcal{O}$ at infinity the curve E. We write $E/\mathbb{R}$ or $E/\mathbb{F}_q$ for a curve E defined over the field $\mathbb{R}$ or $\mathbb{F}_q$.

Definition 12 ($P * Q$). Suppose we are given two points, P and Q on E. We define $P * Q$ on E as follows: We let $P * Q$ be the third point of intersection of the line PQ with E. If $P = Q$, then we take the line to be the tangent to E at P (i.e. P has a multiplicity of 2 or 3).

To better understand this definition, let us look at some examples. Let $P = (x_P, y_P)$, $Q = (x_Q, y_Q)$, $R = (x_R, y_R)$, and $\tilde{P} = (x_P, -y_P)$. We note the following:

1. $\mathcal{O} * \mathcal{O} = \mathcal{O}$, since $\mathcal{O}$ is a point of inflection.
2. $P * \mathcal{O} = \tilde{P}$ and $P * \tilde{P} = \mathcal{O}$ (see Figure 5.2).
3. $P * Q = Q * P$.
4. If $P * Q = R$, then $P * R = Q$ and $Q * R = P$ (see Figure 5.3).
5. If $Q = P$, we take the tangent line (see Figure 5.4). In this figure, note that $P * P = R$ and $P * R = P$.

Definition 13 ($P + Q$). Select a point on E; call it $\hat{\mathcal{O}}$. We define $P + Q = (P * Q) * \hat{\mathcal{O}}$.

We traditionally choose $\hat{\mathcal{O}} = \mathcal{O}$ for curves in Weierstrass canonical form. With this choice, it leads us directly to our next theorem.

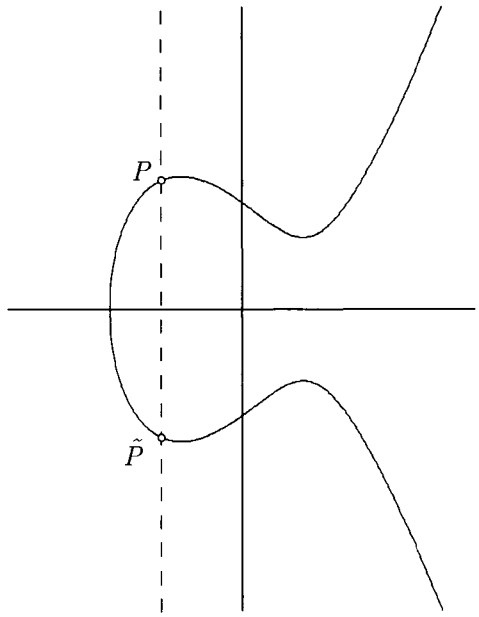

Figure 5.2: $-P$ on an elliptic curve

Theorem 21 (Group Law). *The set of all points on E (including $\mathcal{O}$) defines an abelian group with respect to addition.*

We omit a complete proof. To prove this, we need to show the following:

1. Closure: If $P, Q \in E/\mathbb{R}$, then $P + Q \in E/\mathbb{R}$.

2. Associativity: For all $P, Q, R \in E/\mathbb{R}$, $(P + Q) + R = P + (Q + R)$.

3. Identity: There exists an element $0 \in E/\mathbb{R}$ such that $P + 0 = P$ for all $P \in E/\mathbb{R}$.

4. Inverses: For each $P \in E/\mathbb{R}$, there exists $-P \in E/\mathbb{R}$ such that $P + (-P) = 0$.

5. Commutativity: $P + Q = Q + P$ for all $P, Q \in E/\mathbb{R}$.

Proof. Closure and commutativity are clear (See Figures 5.2-5.4).

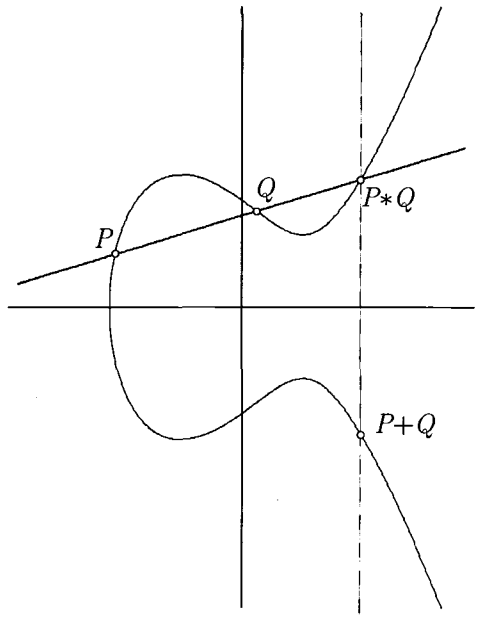

Figure 5.3: Addition on an elliptic curve

- The additive identity (see Figure 5.2) is $\mathcal{O}$, since

$$\begin{aligned} P + \mathcal{O} &= (P * \mathcal{O}) * \mathcal{O} \\ &= \tilde{P} * \mathcal{O} \\ &= P. \end{aligned}$$

- The additive inverse (see Figure 5.2) $-P$ is $\tilde{P}$, since

$$\begin{aligned} P + \tilde{P} &= \left(P * \tilde{P}\right) * \mathcal{O} \\ &= \mathcal{O} * \mathcal{O} \\ &= \mathcal{O}. \end{aligned}$$

From this point forward, we will use $-P = (x_P, -y_P)$ instead of $\tilde{P}$.

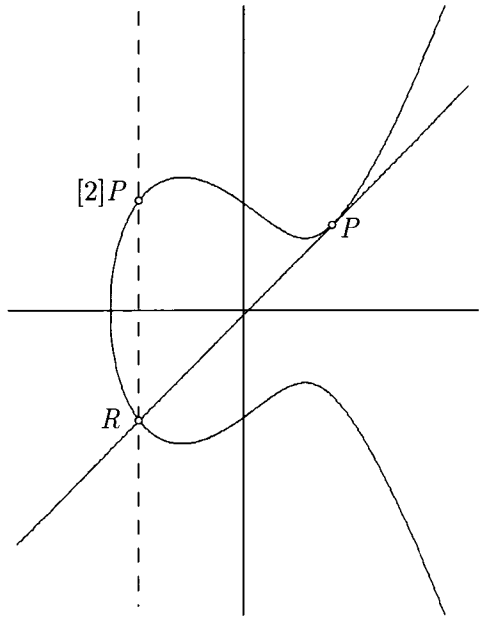

Figure 5.4: Point doubling on an elliptic curve

- Ironically, associativity is the difficult property to prove (See Enge [8, pages 40-41] for a proof.). □

Since this is a group, we now can define point multiplication.

Definition 14 (Point Multiplication)**.** We set

$$[k]\,P = \underbrace{P + P + \cdots + P}_{k \text{ times}} \text{ for } k \in \mathbb{N}.$$

Let us look at addition more closely.

1. Addition of two points when $P \neq Q$:

   (a) If $x_P = x_Q$, then $Q = -P$, so $P + (-P) = \mathcal{O}$ (see Figure 5.2).

   (b) If $x_P \neq x_Q$, let $P + Q = R$ where $R = (x_R, y_R)$ (see Figure 5.3). We calculate R as follows:

i. We first find the slope of the line through the points P and Q:

$$\lambda = \frac{y_Q - y_P}{x_Q - x_P}.$$

ii. Next, we plug

$$y = \lambda (x - x_P) + y_P \tag{5.2}$$

into Equation 5.1. So, we have

$$\begin{aligned} (\lambda (x - x_P) + y_P)^2 &= x^3 + ax + b \\ 0 &= x^3 - x^2\lambda^2 + x\left(2\lambda^2 x_P - 2\lambda y_P + a\right) \\ &\quad -\lambda^2 x_P^2 + 2\lambda x_P y_P - y_P^2 + b. \end{aligned}$$

We know the coefficient of x^2 is the sum of the roots, $\lambda^2 = x_P + x_Q + x_R$. Thus

$$x_R = \lambda^2 - x_P - x_Q.$$

iii. Finally, once we have x_R, we plug it back into Equation 5.2 and change the sign to get

$$y_R = \lambda (x_P - x_R) - y_P.$$

2. Addition of two points when $P = Q$ (i.e. Point Doubling):

(a) If $y_P = 0$, then $P = -P$, so $[2]P = \mathcal{O}$.

(b) If $y_P \neq 0$, let $[2]P = P + P = R$ where $R = (x_R, y_R)$ (see Figure 5.4). We calculate R as follows:

i. The line is the tangent at x_P. We get the slope using implicit differentiation

on Equation 5.1. Thus we get

$$2yy' = 3x^2 + a;$$

so $\lambda = y'$ at (x_P, y_P). Therefore we get

$$\lambda = \frac{3x_P^2 + a}{2y_P}.$$

ii. As before, λ^2 is the sum of the roots, but x_P is a root with multiplicity 2, so

$$x_R = \lambda^2 - 2x_P.$$

iii. Finally, once we have x_R, we plug it and λ into Equation 5.2 to get

$$y_R = \lambda(x_P - x_R) - y_P.$$

Using these formulas, one can prove associativity but it is a long algebraic mess.

What is particularly nice about addition on E is that it is entirely algebraic. Hence, these formulas hold even if we change the field. In particular, let us look at E over $\mathbb{F}_q$ (Note that we have to calculate different algebraic formulas for $P + Q$ if $q = 2^r$ or 3^r; this is why we avoid this case).

Elliptic Curves over $\mathbb{F}_q$

Though addition on E is defined geometrically (over $\mathbb{R}$), the formulas are algebraic, thus they work over any field. To have a practical use in cryptography, we limit our field to $\mathbb{F}_q$ where $q = p^r$ is large, p is prime, and $r \in \mathbb{N}$. When $p = 2$ or 3, the change of variables used

to get the canonical form in Equation 5.3 includes division by zero (and, as noted earlier, we have to calculate different algebraic formulas). Over these fields, the canonical forms are:

1. $y^2 + xy = x^3 + ax^2 + b$, if $p = 2$ and
2. $y^2 = x^3 + ax^2 + bx + c$, if $p = 3$.

There are also constraints on a and b similar to our earlier constraint $4a^3 - 27b^2 \neq 0$ [12]. Although the case of $q = 2^r$ is common because it lends itself nicely to binary, our attention is on when p is a large prime and $r = 1$. Thus, from this point forward we assume that our field, $\mathbb{F}_p$, satisfies the condition that p is a large prime ($r = 1$). Therefore with minor adjustments to equation 5.1 we have:

$$y^2 \equiv x^3 + ax + b \mod p \tag{5.3}$$

where $a, b, x, y \in \mathbb{F}_p$ and $4a^3 + 27b^2 \not\equiv 0 \mod p$. Let us look at an elementary example over $\mathbb{F}_{23}$ with $a = 1$ and $b = 2$.

1. Note $4a^3 + 27b^2 \equiv 4 \cdot 1^3 + 27 \cdot 2^2 \equiv 4 + 108 \equiv 112 \equiv 20 \not\equiv 0 \mod 23$.
2. Note that $P = (0, 5)$, $Q = (8, 4)$, $R = (0, 18)$, and $S = (10, 0)$ all lie on E. Let us find the following:
   (a) $P + R$. Because $x_P = 0 = x_R$ we have:

$$P + R = \mathcal{O}.$$

   (b) $P + Q$. Because $x_P = 0 \neq x_Q = 8$ we have:

i. $\lambda \equiv \frac{y_Q - y_P}{x_Q - x_P} \mod p$:

$$\lambda \equiv \frac{4-5}{8-0} \mod 23$$

$$\lambda \equiv 22 \cdot 8^{-1} \mod 23$$

$$\lambda \equiv 22 \cdot 3 \mod 23$$

$$\lambda \equiv 20 \mod 23$$

ii. $x_3 \equiv \lambda^2 - x_P - x_Q \mod p$:

$$x_3 \equiv 20^2 - 0 - 8 \mod 23$$

$$x_3 \equiv 400 - 8 \mod 23$$

$$x_3 \equiv 1 \mod 23$$

iii. $y_3 \equiv \lambda (x_P - x_3) - y_P \mod p$:

$$y_3 \equiv 20 \cdot (0-1) - 5 \mod 23$$

$$y_3 \equiv (20 \cdot -1) - 5 \mod 23$$

$$y_3 \equiv -25 \mod 23$$

$$y_3 \equiv 21 \mod 23$$

Thus $P + Q = (1, 21)$.

(c) $[2]\,S$. Because $y_S = 0$ we have:

$$[2]\,S = \mathcal{O}.$$

(d) $[2]\,P$. Because $y_P \neq 0$ we have:

i. $\lambda \equiv \frac{3x_P^2 + a}{2y_P} \mod p$:

$$\lambda \equiv \left(3 \cdot 0^2 + 1\right) \cdot (2 \cdot 5)^{-1} \mod 23$$

$$\lambda \equiv 1 \cdot 10^{-1} \mod 23$$

$$\lambda \equiv 1 \cdot 7 \mod 23$$

$$\lambda \equiv 7 \mod 23$$

ii. $x_2 \equiv \lambda^2 - 2x_P \mod p$:

$$x_2 \equiv 7^2 - 0 \cdot 1 \mod 23$$

$$x_2 \equiv 49 \mod 23$$

$$x_2 \equiv 3 \mod 23$$

iii. $y_2 \equiv \lambda\left(x_P - x_2\right) - y_P \mod p$:

$$y_2 \equiv 7 \cdot (0 - 3) - 5 \mod 23$$

$$y_2 \equiv (7 \cdot -3) - 5 \mod 23$$

$$y_2 \equiv -26 \mod 23$$

$$y_2 \equiv 20 \mod 23$$

Thus $[2]\, P = (3, 20)$.

Security of ECC over $\mathbb{F}_p$

As we stated before, ECC basically uses the same idea as ElG. The security of ECC lies in the discrete log problem. It is easy for us to calculate $[k]\,P = Q$ given k and P, but extremely difficult to calculate k given P and Q. When we use ECC there are parameters that must be agreed upon before starting; they are $\mathbb{F}_p$, the elliptic curve to be used (i.e. a and b such that $4a^3 + 27b^2 \not\equiv 0 \mod p$), and a base point $P \in E/\mathbb{F}_p$. To ensure the security, we choose p, a prime between 50-100 digits. Since these parameters do not have to be random and can be checked by others, they can be agreed upon in advance. Let us look at a simplified demonstration of how ECC works.

1. Alice and Bob agree on p, $\mathbb{F}_p$, $y^2 \equiv x^3 + ax + b \mod p$ and $P \in E/\mathbb{F}_p$.
2. Alice chooses a random integer u such that $1 \leq u \leq p-1$.
   (a) She computes $[u]\,P = Q$.
   (b) She makes Q public and keeps u secret.
3. Bob chooses a random integer v such that $1 \leq v \leq p-1$.
   (a) He computes $[v]\,P = R$.
   (b) He makes R public and keeps v secret.
4. Alice and Bob compute $S = [uv]\,P$, their shared secret key to be used in an existing symmetric-key cryptographic system.

Each $E/\mathbb{F}_p$ generates a number of points, call it N. Although we do not need to know the value of N to implement our system, to ensure the security of it, we choose an $E/\mathbb{F}_p$ such that N is not a product of small primes. Various methods exist for computing N (See Koblitz for techniques [12, page 183]), however we are assuming that our N is not a product

of small primes. The next critical step in our system is the choice of P. As long as $P \neq \mathcal{O}$, it will generate a subgroup of $E/\mathbb{F}_p$. We choose P such that the order is almost as large as N. To ensure that our choice of P is acceptable, we choose $E/\mathbb{F}_p$ such that N is a prime [12, page 184]. Thus by our choices we have made the elliptic curve discrete log problem the most formidable and labor intensive making our system secure.

All the public-key ciphers we have studied are considered hard to break. What makes ECC attractive is the relatively small key size (i.e. the complexity to break the cipher). In Chapter 4, we studied ElG and the difficulty of the discrete log problem. Like ElG, ECC is defined over a finite field. With the discrete log problem over a finite field we were able to define the field with respect to addition and multiplication. With the elliptic curve discrete log problem over a finite field we only defined a group with respect to addition. This difference between the two problems is what makes ECC more arduous [8, pages 118-124].

CHAPTER 6

CONCLUSION

Although our discussion has focused on asymmetric-key cryptography, it is often not used as the sole source of encryption for communication. Public-key cryptography is generally used in a cipher suite. A cipher suite consists of:

1. Asymmetric-key cryptographic systems used for key-exchange protocol and digital signatures. (e.g. RSA, ElG, ECC)

2. Symmetric-key cryptographic systems used for secure communications. (e.g. AES, Advance Encryption Standard; DES, Data Encryption Standard)

3. One-way hash functions used for connection integrity. (e.g. SHA, Secure Hash Algorithm; MD5, Message Digest algorithm)

A cipher suite is commonly used for secure communications over the internet in the handshake protocol. The protocol can be summarized in the following steps:

1. ClientHello: A user connects to a server presenting a list of cryptographic systems supported by the user's system.

2. ServerHello: The server chooses the cryptographic systems to be used and informs the user of the decision.

3. ServerCertificate: The server sends its digital certificate to the user certified by a trusted third party.

4. ClientKeyExchange: The user chooses a random number (premaster secret), encrypts it with the server's public-key, and sends it to the server. Both the user and server generate the master secret from the premaster secret.

5. ChangeCipherSpec: User (server) informs the server (user) everything will be encrypted from this point forward.

6. Finish: User (server) sends hash function for verification.

We have studied three asymmetric cryptographic systems and the mathematics that make them work. Currently first generation ciphers require the generation of very large secret keys to ensure the security of the information. The next generation of public-key ciphers, elliptic curve cryptography, dramatically reduced the size of the secret key as well as increased the security level. Although cracking ECC keys is considered infeasible for large primes, we cannot guarantee how long this will last. Recently a company in Canada, D-Wave, produced a working quantum computer albeit only 16-qubit. This model is no faster than today's digital computers and still lacks the efficiency to perform discrete logarithms and factoring large numbers. However, D-Wave plans to have a 1024-qubit computer operational by the end of 2008 [7]. This very well could turn the world of public-key cryptography upside down. If this were to happen, we would need to speed up the research into quantum cryptography. Quantum computing and quantum cryptography share the same basis of laws in quantum mechanics but, quantum cryptography does not require quantum computing. If history has taught us one thing about secret codes, it is once one is implemented there is someone out there trying to break it.

BIBLIOGRAPHY

[1] Abrams, Marc, *World Wide Web: Beyond the Basics*, Pearson Prentice Hall, Upper Saddle River, NY (1998).

[2] Anderson, James, James Bell, *Number Theory with Applications*, Pearson Prentice Hall, Upper Saddle River, NY (1997).

[3] Blake, Ian, Gadiel Seroussi, Nigel Smart, *Elliptic Curves in Cryptography*, Cambridge: Cambridge University Press (1999).

[4] Burton, David, *Elementary Number Theory*, 6th Edition, McGraw-Hill, New York, NY (2007).

[5] Certicom, *Online Elliptic Curve Cryptography Tutorial*, (2007), $< http : //www.certicom.com/index.php?action = ecctutorial, home >$, (08/01/2007).

[6] Crandall, Richard, Carl Pomerance, *Prime Numbers: A Computational Perspective*, 2nd Edition, Springer, New York, NY (2005).

[7] D-Wave Systems, *News*, (1999), $< http : //www.dwavesys.com/ >$, (07/01/2007).

[8] Enge, Andreas, *Elliptic Curves and Their Applications to Cryptography: An Introduction*, Kluwer Academic Publishers, Norwell, NJ (1999).

[9] Goldreich, Oded, *Foundations of Cryptography: Basic Tools*, Cambridge University Press, Cambridge, UK (2001).

[10] Haufler, Hervie, *Codebreakers' Victory: How the Allied Cryptographers Won World War II*, New American Library, New York, NY (2003).

[11] Kahn, David, *The Codebreakers: The Story of Secret Writing*, Scribner, New York, NY (1996).

[12] Koblitz, Neal, *A Course in Number Theory and Cryptography*, 2nd Edition, Springer, New York, NY (1994).

[13] Koblitz, Neal, *Introduction to Elliptic Curves and Modular Forms*, Springer, New York, NY (1984).

[14] Koshy, Thomas, *Elementary Number Theory with Applications*, 2nd Edition, Elsevier Inc., Burlington, MA (2007).

[15] Miller, Steven, Ramin Takloo-Bighash, *An Invitation to Modern Number Theory*, Princeton University Press, Princeton, NJ (2006).

[16] Mollin, Richard, *RSA and Public-key Cryptography*, CRC Press, Boca Raton, FL (2003).

[17] Nathanson, Melvyn, *Elementary Methods in Number Theory*, Springer, New York, NY (2000).

[18] Pincock, Stephen, *Codebreaker: The History of Codes and Ciphers*, Walker and Company, New York, NY (2006).

[19] RSA Laboratories, *RSA Laboratories Frequently Asked Questions About Todays Cryptography, Version 4.1*, (2000), $< http://www.rsa.com/rsalabs/node.asp?id = 2152 >$, (06/01/2007).

[20] Scheinerman, Edward, *Mathematics: A Discrete Introduction*, Brooks Cole, Pacific Grove, CA (2000).

[21] Schneier, Bruce,*Applied Cryptography*, John Wiley and Sons, Inc., New York, NY, (1996).

[22] Secure Trust, *History of Cryptography*, (2004), $< https://www.securetrust.com/resources/cryptography - history/ >$. (06/01/2007).

[23] Spillman, Richard, *Classical and Contemporary Cryptology*, Pearson Prentice Hall, Upper Saddle River, NJ, (2005).

[24] US-CERT, *Vulnerability Notes Database*, (2003), $< http://www.kb.cert.org/vuls/bypublic >$. (07/01/2007).

VITA

Graduate College
University of Nevada, Las Vegas

Thomas Hodge

Degrees:

Associate of Science, Medical Laboratory Medicine, 1996
Community College of the Air Force, Montgomery, Alabama

Associate of Science, Communications, 1998
Community College of the Air Force, Montgomery, Alabama

Bachelor of Science, Mathematics, 2002
Angelo State University, San Angelo, Texas

Thesis Title: Cryptography in the Digital Age

Thesis Examination Committee:

Chairperson, Dr. Arthur Baragar, Ph.D.
Committee Member, Dr. David Costa, Ph.D.
Committee Member, Dr. Peter Shiue, Ph.D.
Graduate Faculty Representative, Dr. Yoohwan Kim, Ph.D.