ELLIPTIC CURVE CRYPTOGRAPHY

by

Dave Thompson

Submitted in partial fulfillment of the

requirements for Departmental Honors in

the Department of Mathematics

Texas Christian University

Fort Worth, Texas

May 2, 2016

ELLIPTIC CURVE CRYPTOGRAPHY

Project Approved:

Supervising Professor: Qiao Zhang, Ph.D.

Department of Mathematics

Liran Ma, Ph.D.

Department of Computer Science

Ze-Li Dou, Ph.D.

Department of Mathematics

ABSTRACT

While it is relatively well known that larger field orders in elliptic curves allow for increased security in a cryptographic setting, it was the goal of our research to discover if patterns would emerge when observing average point orders within the first five fields. Using SageMath, we computed the average point orders across every elliptic curve with constraints subject to a given finite field that passed through each point. From this, we also calculated an average of averages that we used to represent the average point order for the entire field. Our findings point towards the fact that if patterns do exist, they are complex and ultimately require more data sets than those which have been analyzed in this report. While our data suggests an almost linear relationship between the increase in field size with the increase in average point order, in truth this relationship may deviate significantly from one that is linear after larger data sets are taken into account.

Contents

CHAPTER 1

An Introduction to Symmetric Key Cryptography

1. Introduction

Cryptography, or cryptology, is the study of security as it relates to communication. Specifically, it consists of two main processes: encryption and decryption. Encryption is the process by which a message is altered so as to disguise its meaning from a third party. Similarly, decryption is the process by which the altered message is reverted back to the original message, thereby allowing the its meaning to become clear. The way in which these processes are carried out is through a cipher, which is the algorithmic instruction for performing the encryption and decryption (in the reverse form).

In terms of structure, there are two different types of key systems: symmetric key and asymmetric key systems. Until 1976, the only type of systems that had been discovered were symmetric key systems. In symmetric key systems, it is necessary for both parties to have knowledge of the particular cipher. In contrast, asymmetric key systems do not require that both parties have knowledge of the same cipher. The importance of the cipher in a symmetric key system is clear from the example that follows.

2. Symmetric Key Example

Suppose person A wishes to send a message to person B, without person C being able to understand the message. Suppose the original message is "XYZ" and the cipher is A maps to Z, B maps to Y, C maps to X, etc. Then we see that if person A wants to encrypt the message "XYZ", he/she will send the encrypted message "CBA". However, if only person A has access to the cipher, person B will be unable to decrypt the message, as "CBA"" could in fact mean any of $26 \times 25 \times 24 = 15,600$ different messages, which would be even larger if more letters were used in the original message or if extra characters were allowed. That means that without person B knowing the cipher, the probability that he/she would guess the original message correctly would be $1/15{,}600 \approx .0000641$. To ensure that person B is able to understand the meaning of the original message, it is clear the receiving party must know the decryption cipher.

Similarly, if person A is wanting to send the same message "XYZ", but does not know the encryption cipher, he/she must either randomly assign new values for each letter, or send the original message to person B. In either case, person A fares a poor chance of successfully communicating the original message with person B, as ultimately that "encrypted" message will be decrypted according to a different procedure[1], thereby inaccurately stating the original message. Thus,

[1] except for the one time in 15,600 where person A is able to correctly guess the cipher

if person A does not know the cipher, it is clear that person B would have little hope of discovering the original message. From this example, it is clear that both parties must have an agreed upon cipher prior to sending encrypted messages in communication channels.

3. Some Commonly Applied Symmetric Key Cryptographic Systems

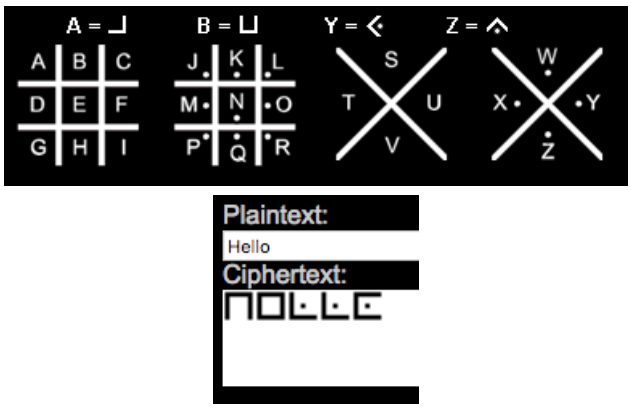
3.1. Introduction. Although computers have been instrumental in the development of encryption and decryption algorithms, also called ciphers, as well as overall information security, encrypting information is a practice that was adopted long before messages were even able to be sent electronically.

3.2. Caesar Cipher. During time periods where messages were required to be physically delivered, the ciphers that were applied to the sent message were more rudimentary, generally consisting of a bijective mapping from one letter of the alphabet to another, similar to the example described earlier. These types of ciphers are called Caesar ciphers, as these were used by Julius Caesar during ancient Roman times. A particular Caesar cipher structure is generally denoted as n-cipher where $A \to n$ (for some letter of the alphabet, n), $B \to n+1$ (the letter that follows n, unless n is Z, in which case $n+1$ is equal to A), etc. For example, G-cipher denotes the cipher where $A \to G$, $B \to H$, etc.

Whereas today these types of ciphers can be solved extremely quickly with the help of computers, during that era these types of ciphers would have been adequate in encrypting the message to the extent that solving a particular cipher could require hours of trial and error from a third party, especially given that there were 25 possible ciphers.[2]

3.3. Pigpen Cipher. Another type of cipher that was used in times before electronic communication was the pigpen cipher, sometimes called the Freemason cipher, as it was made famous by its practice among Freemason members. In addition, there is evidence that suggests that the Knight"s Templar utilized a pigpen cipher in their communications as well. With the pigpen cipher, letters of the alphabet are represented by families of symbols as indicated by a key. An example of such a key with a plain text and encrypted text can be found below.

The following example is taken from [**4**].



[2]Here we assume that the identity cipher would not be used.

3.4. Book Cipher. A particularly unique type of cipher that was also used in pre-electronic communications was the book cipher. In contrast with ciphers that map a symbol or letter to another letter in the decryption process, a book cipher utilizes a book or other piece of literature as its key. To encrypt a word or letter, one may simply reference to a specific word or letter in the piece of literature that both parties have in their possession. Historically, the bible was popularly utilized for this cipher, as it was readily available to most individuals.

For example, if the fifth word in the third paragraph of the twelfth page is ”apple”, one could encrypt apple by simply writing ”12, 3, 5”. Although this would be seemingly very difficult to decrypt without access to or knowledge of the referenced literature, in practice it is quite possible for the cipher to be solved without even a knowledge of which reference literature is used, especially if identical encrypted words are not represented by different references. Thus, although the book cipher differs from other ciphers in its structure, it can be analyzed using similar techniques as with other ciphers, making the strength of the cipher dependent not on the choice of literature but only on the complexity of the message itself.

3.5. Grille Cipher. Lastly, a popular type of cipher used in pre-electronic communications, made famous during the colonial era in United States history, is the grille cipher, which is still used as a form steganography today, albeit in a much different form. A grille cipher consisted generally of a mask that was placed on top of an encrypted message to reveal the true message.

For example, in an encrypted paragraph, a particular mask may cover up 3 different words of each row, revealing a new paragraph that is the intended message. While this cipher can in many ways be more difficult to solve than Caesar ciphers or other letter based ciphers, it also has more degrees of freedom in that it relies on the skill of the individual performing the encryption to either disguise the fact that any sort of encryption has occurred or to include enough distractions so as to complicate the decryption process for any third party.

The strength of this process is that, if done well, the third party may not realize that an encryption process has occurred. Whereas in other encryption methods there may be an obvious need for the message to be decrypted, grille ciphers can give the illusion that no such secret message exists, posing a problem for third parties that are looking for specific information.

4. The Limitations of Symmetric Key Cryptographic Systems

After the introduction of computers, different cryptographic schemes emerged, leaving some previously favored symmetric key ciphers to the wayside. Moreover, from improvements in computing power, a cipher that may have at one time taken an individual days or even weeks to solve could often be solved by a computer in a matter of minutes. For this reason, the growing computing power available to the public has greatly reduced the viability of many of the aforementioned ciphers in modern communication systems.

Additionally, while symmetric key cryptographic systems generally present simpler computations and therefore faster message exchanges, there is also an increased risk of a breach, as compared to asymmetric key systems. This is because the encryption function is directly reversible, in contrast to the asymmetric key structure. By using a reversible function to encrypt a message, a third party that

gains access to the key can gain access to messages sent by either party. As we will see in the next chapter, in the asymmetric key case, gaining access to one private key only guarantees access to messages in one direction.

CHAPTER 2

An Introduction to Asymmetric Key Cryptography

1. Introduction to the Public Key System

One of the key systems that emerged with the rise of computers was the public key system. In 1976, the first example of an asymmetric key system was published. This new system, named the public key system, was unique in that it was the first type of cryptographic system to not follow the symmetric key structure. Earlier, it was demonstrated that in a symmetric key structure, it is necessary for both parties to have knowledge of the particular cipher. This is not the case with asymmetric key structures and consequently not the case with the public key system.

While there are many different types of public key systems, each system operates in a similar fashion. Specifically, in the public key system, each public key is paired with a private key, of which there exists a pair for each party. Thus, if person A wishes to communicate with person B, then we will have a total of four keys: public key A, private key A, public key B, and private key B.[1] As we will see later, although four keys are generated, during a one-direction message transfer only two of the four keys are utilized for a given direction, as the public and private keys of the party sending the message is not used unless a return message is sent.

In the example that follows, we will generate two keys (public key B and private key B) as well as illustrate a one-direction message transfer between two parties, person A and person B. A third party, person C, is able to see public key A and public key B but is unable to see private key A and private key B. In this system, encryption is done through the public key, meaning anyone, even a third party, is able to encrypt a message. Specifically, if person A is wanting to send a message to person B, he/she will utilize public key B, that is, the public key of the recipient, during the encryption process. In contrast, the decryption process is done through the private key, meaning only the individual with the private key is able to see the intended message. Moreover, the keys are constructed in such a way that one cannot definitively deduce the private key from the public key.

To ensure that one cannot definitively deduce the private key from the public key, ciphers of the various different types of public key systems rely on mathematical calculations that are relatively difficult to solve or that require significant computing power. For example, many public key systems utilize the difficulty of testing for primality or solving a discrete logarithm problem.

The first two public key systems that emerged were the RSA (an acronym for its founders, Rivest, Shamir, and Adleman) and Diffie-Hellman encryption systems. Because of the general similarities between the two encryption schemes, we

[1] In general, for an n-person communication scheme, we would have a total of n public keys and n private keys, resulting in a total of $2n$ keys.

will demonstrate how the encryption and decryption process works for the public key cryptography system by using an RSA cipher example in Section 3 of this Chapter.

2. An Introduction to RSA

RSA is one of the most popular forms of public key cryptography, and its security relies on the difficulty of systematically constructing the prime factorization of numbers. We will highlight the power of RSA encryption through a simplified example, utilizing small primes that are easier to work with than those that are typically used in a more practical setting. Recall that although one could generate all four keys, in the following example, it is not necessary to have four keys since we will eventually use a one-direction communication structure.[2]

3. An Asymmetric Key Example in RSA

Suppose person A wishes to communicate with person B, then, as described earlier, we will have a total of four keys: public key A, private key A, public key B, and private key B. For our purposes, we will limit this example to the construction of person B's set of keys. To construct public key B, person B's computer will generate a pair of prime numbers. As explained earlier, generally each pair of primes consist of very large prime numbers, so as to complicate the practice of a brute force approach in code breaking; however, in this example we will utilize prime numbers that are relatively small. Suppose computer B generates the prime pair (7,11).[3]

For the sake of notation, we define each of the primes using the following assignment: $p_B = 7$, and $q_B = 11$.[4] From these, we can construct the modulus of public key B by taking the product p_B and q_B. Then, for public key B, the modulus is $7 \times 11 = 77$.[5] These modulus values will serve as instrumental to increasing the difficulty of finding a way to deduce the private information.[6] Without extra information, choosing the right solution to a modulus equation would prove to be an impossible task. As such, we will need to include an additional piece of information in our public key.

The second piece of information is generated in two steps. The first of these steps is computing the function $f(p,q) = (p-1) \times (q-1)$. Then, we see that $f(p_B, q_B) = 60$.[7] From here, the RSA algorithm requires that we choose a number that is relatively prime to $f(p,q)$. In our example, we may choose from the set {7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59} for $f(p_B, q_B)$.[8]

---

[2]That is, we will ignore the case where person B responds since it will mirror the process of person A sending a message.

[3]We can let, for example, computer A generate the prime pair (3,5).

[4]Similarly, we can define $p_A = 3$, $q_A = 5$.

[5]For public key A, we see that the modulus is $3 \times 5 = 15$.

[6]A modulus value affects how the arithmetic operations are defined. For example, in modulus 4, 3 + 2 = 5 is congruent to 1, since 5 mod 4 = 1. In general, in modulus 4, we see that $4n + 1$ mod 4 = 1. From this we see that for a given equation, in modulus m, there are potentially infinite congruent solutions.

[7]Similarly, we see that $f(p_A, q_A) = 8$.

[8]Similarly, we may choose from the set {3, 5, 7} for $f(p_A, q_A)$.

We see that the second set is clearly more complicated and, as one might expect, would present a much more difficult challenge of being compromised. Suppose person A chooses 5 from the set and person B chooses 17. Then, we can represent public key B as public key B = (17,77).[9] Note that this is the only information that is made public, as the two prime numbers used to make public key B will remain undisclosed. Now that each party has created his/her public key, we have developed a system to encrypt messages. Specifically, for person A wanting to send a message to person B, he/she will be able to do so through the use of public key B, or (17,77).[10] The problem is that we will need to develop a system for person B to decrypt messages; that is, we require the creation of private key B.

From the public key, the corresponding private key is relatively straightforward to compute. As with the public key, the private key is written as a pair (decryption exponent, modulus). As one might expect, the modulus values is the same as the corresponding public key, meaning all that remains is the computation of the decryption exponent. Since the modulus for public key B is 77, we see that the modulus for private key B is also 77. Thus all that remains is to calculate the decryption exponent, y, for private key B. To do so, we must solve the equation $17y \bmod 60 = 1$. After a quick trial and error, we see that $17 \times 7 \bmod 60$ = -1, so we would expect that $y = 60 - 7 = 53$ to be a solution. Indeed, we can verify that $17 \times 53 = 901 = 1$ in modulus 60. Hence, we see that private key B = (53, 77).[11]

Now that we have defined both of the keys needed for RSA encryption and decryption, we can demonstrate the way in which a typical message is encrypted and decrypted. While typically most messages will not be limited to a series of numbers, in practice messages are converted to numbers usually by some form of a concatenation of an alphabetical mapping from letters to their order in the alphabet. For example, the word "cat" may map to 312. For longer messages, a space or other syntactical expressions may be mapped to an integer as well. Suppose person A wishes to send the message "B" to person B. We can represent "B" as the number 2 and initiate the encryption and decryption process.[12]

Since person A is the individual sending the message, he/she will utilize public key B in encrypting the message. As we have seen, since public key B is in fact public, it is true that anyone would be able to encrypt a message that wants to send person B a particular message. However, it is also true that a third party would not have access to the private key, and therefore we can ensure that any message

[9]For public key A, we would have public key A = (7,15).

[10]For n parties wishing to communicate with a single receiver, it is only necessary to have one public key. More specifically, the public key that is only required during a message transfer is the public key of the receiving party. For this reason, we have restricted our computations of the keys for person A to footnotes.

[11]To calculate our decryption exponent, x, for private key A, we need to solve the equation $7x \bmod 8 = 1$. Note that 8 comes from our first step in computing the original modulus. Although this equation is straightforward, in truth it may be tedious to solve, especially as the original primes become much larger. A quick trial and error reveals that $x = 1$ leads to $7 \bmod 8 = -1$, and therefore we would expect a solution to be $8 - 1 = 7$. We can verify this with $7 \times 7 = 49$ which is congruent to 1 in modulus 8. Thus, we conclude that private key A = (7, 77).

[12]This process of assigning a numerical value to a message is generally done by a hash function, a designated function that converts message characters to numbers prior to encryption. While sometimes this function converts a word or phrase directly into a number, more often than not the hash function serves as a concatenation of function values for each character or word from a given message needing to be encrypted.

sent to B will only be able to be decrypted by person B. From earlier, we recall that public key B = $(17, 77)$, and private key B = $(53, 77)$. Using public key B, person A can encrypt the message 2 by encrypting it into a cipher text. In RSA, the cipher text is computed by taking the plain text raised to the power of the encryption exponent, the first in the pair of the public key, in the public key modulus. Thus, for person A wanting to encrypt 2, he/she can do so as follows: cipher text = 217 mod 77. After computing, we see that the cipher text is equal to 18. Therefore, after encryption we see that person A's message has changed from 2 to 18. To ensure that person B can read the intended message, he/she will need to decrypt the message using the private key. The process for decrypting a message using the private key mirrors that of the process for encryption using the public key. In this example, person B can decrypt the cipher text as follows: plaintext is equal to 1853 mod 77. After computing we see that the plaintext is equal to 2, which was the intended message of person A.

4. The Limitations of Asymmetric Key Cryptographic Systems

The main limitation of asymmetric key cryptographic systems is that they require significant more time and computation than symmetric key cryptographic systems. This means that some weaker processors, such as those found on older phones and computers, may have difficulty running such cryptographic systems, as the large computational demands may fail to work quickly on such devices. Over the last 10 years, standard devices, especially mobile platforms, have become much more powerful in terms of processing capacity, and therefore have been able to handle such encryption schemes.

Another limitation of asymmetric key systems is the fact that they can still be exploited by a variety of hacking methods. Thus, the security of these systems is certainly not guaranteed, and over time, as computing speed improves, these methods will become more and more viable in terms of their success in hacking these systems quickly. Thus, more advanced methods will be needed to combat the rise in computing power.

CHAPTER 3

An Introduction to Elliptic Curves

1. Introduction

For a finite field, $\mathbb{F}_q$, of order greater than 3, we define an elliptic curve, E, to be the graph of the Weierstrass equation, defined by

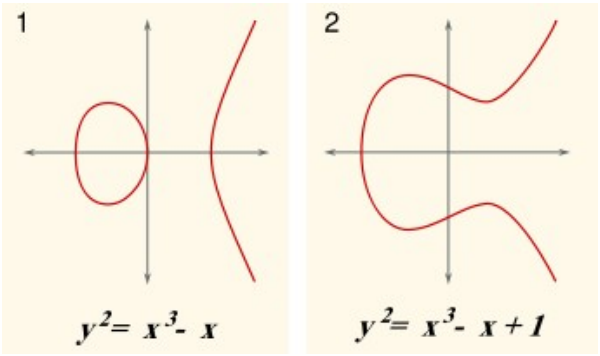
$$y^2 = x^3 + Ax + B, \tag{1}$$

where A and B are constants, and A, B, x, and y are elements of the field $\mathbb{F}_q$. For a finite field of order 2 of 3, we define the elliptic curve, E, to be the graph of the generalized Weierstrass equation, defined by

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6. \tag{2}$$

Further, we restrict our definition to include only those curves that are non-singular. This is satisfied by assuming that $4A^3 + 27B^2$ is not equal to 0.

Two examples of elliptic curves can be seen below.



2. The Point at Infinity

To give elliptic curves their group structure, it is necessary to introduce the "point at infinity".[1] This is done by utilizing projective coordinates for our points on the given elliptic curve. For a given field, $\mathbb{F}$, we define the two dimensional projective space, $\mathbb{P}^2{}_{\mathbb{F}}$, as the set of equivalence classes of triples (x, y, z) with x, y, $z \in \mathbb{F}$, and at least one of the elements being non-zero. We denote the equivalence class of triples (x, y, z) as $(x : y : z)$. We can organize the set of equivalence classes into two categories: those where $z = 0$ and $z = 1$, the ladder of which representing the set of "finite" points in our projective space, and the former representing the "points at infinity". We refer to the set of all points with finite order as the torsion subgroup of E, as it is in fact a subgroup of E. Similarly, the set of all points with

[1] Often this point is denoted by $P = \infty$.

infinite order, meaning kP is never equal to ∞, is called the non-torsion subgroup of E.

Although we will carry out our computations using only the affine coordinates, in truth we can define each point in terms of its projective coordinates. Then, in the two dimensional projective space, we see that (1) becomes

$$y^2z = x^3 + Axz^2 + Bz^3. \tag{3}$$

3. The Group Operation

3.1. Point Addition. In this section, we will define the point addition, which serves as the group operation over a finite field, $\mathbb{F}_q$. We denote the group as $E(\mathbb{F}_q)$.[2]

Given a point $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$, we define the point addition as $P_1 + P_2 = P_3$, where $P_3 = (x_3, y_3)$, and P_3 is the reflection across the x-axis of the point intersecting the elliptic curve and the line secant to P_1 and P_2. Geometrically, the construction is straightforward; however, the arithmetic computation is generally less obvious. It can be shown that for P_1 not equal to P_2 and with secant slope $m = (y_2 - y_1)/(x_2 - x_1)$, we have $x_3 = m^2 - x_1 - x_2$, and $y_3 = m(x_1 - x_3) - y_1$. This gives the final result that $P_3 = (m^2 - x_1 - x_2, mx_1 - mx_3 - y_1)$.

3.2. Doubling a Point. If P_1 is equal to P_2, we have $P_1 + P_2 = 2P_1$, and we can define the doubling of a point using the same general formula as point addition, with the exception that now m represents the slope of the tangent line to the elliptic curve at point P_1. We can solve for this slope, m, by solving $m = (3{x_1}^2 + A)/(2y_1)$. To generalize further, we define the scalar multiplication of a point as $kP = P + P + P + ... + P$ (k times). An algorithm for scalar multiplication of a point is given in the following subsection.

3.3. Scalar Multiplication of a Point Algorithm. This algorithm is taken from [**1**].

For a given positive integer, k, and a point, P, on an elliptic curve, E, we outline the following algorithm to compute the scalar multiplication of a point, kP.

1. Start with $a = k$, B = infinity, $C = P$.
2. If a is even, let $a = a/2$, and let $B = B$, $C = 2C$.
3. If a is odd, let $a = a - 1$, and let $B = B + C$, and $C = C$.
4. If a is not equal to 0, go back to step 2.
5. Output B.

This B is the value of kP.

This algorithm will be very useful in our understanding of the order of a point on an elliptic curve, and it will also serve to demonstrate the difficulty of solving the discrete logarithm problem, as we will observe in the sections that follow.

[2]It is worth noting that the identity element of the group is the point at infinity, meaning the sum of any point, P, and the point at infinity is equal to P. Additionally, the order of the group, $E(\mathbb{F}_q)$, is equal to the number of points on the elliptic curve defined over the finite field.

3.4. Example 1: Adding Points. Let $\mathbb{F} = \mathbb{F}_7$, and let E be defined by $y^2 = x^3 - x$. Suppose we have points P and Q on E, defined by (1,0) and (4,2) respectively. Suppose we want to find the sum $R = P + Q$. We can first verify that both points P and Q lie on E by plugging in the points to our elliptic curve equation. First, we see $1^3 - 1 \bmod 7 = 0$, demonstrating that P lies on E. Next we check that $4^3 - 4 \bmod 7 = 60 \bmod 7 = 4 = y^2$, so $y = 2$, demonstrating that Q lies on E.

To find R, we can use the point addition formula after solving for m. We find m from simply finding the slope of the secant line. Solving then for m, we have $m = (2-0)/(4-1) = 2/3$. Then we have $R = (4/9 - 4 - 1, 2/3(1 - (x_3) - 0)) = (19/9, 106/27)$.

3.5. Example 2: Scalar Multiplication of a Point. Let $\mathbb{F} = \mathbb{F}_{11}$, and let E be defined by $y^2 = x^3 + 2x - 1$. Suppose we have a point, P, on E, defined by (4,7). Suppose we want to find the scalar product $4P$. We can first verify that point P lies on E by plugging in the point to our elliptic curve equation. We see that $4^3 + 2 \times 4 - 1 = 64 + 8 - 1 \bmod 11 = 5 = y^2$, so $y = 7$ ($49 \bmod 11 = 5$), demonstrating that P lies on E.

To find $4P$, we can use the algorithm outlined in the scalar multiplication subsection. We start with $a = 4$, B = infinity, and $C = (4,7)$. Since a is even, we have $a = a/2$, $B = B$, and $C = 2C$. This gives us $a = 2$, $B = B$, and $C = (7,9)$. Since a is even again, we have $a = a/2$, $B = B$, and $C = 2C$. This gives us $a = 1$, $B = B$, and $C = (2,0)$. Lastly, since a is odd, we have $a = a - 1$, $B = B + C$, and $C = C$. This gives us $a = 0$, $B = (2,0)$, $C = (2,0)$. Outputting B, we have $4P = (2,0)$.

4. The Order of the Group

For elliptic curves over a finite field, the order of the group is equal to the number of points on the elliptic curve. While for small fields, it may not seem time-consuming to count the number of points on the elliptic curve, in practice, many elliptic curves are chosen with exceedingly large orders, necessitating the use of a faster process. While early on, algorithms such as the Baby Step, Giant Step method were able to be applied to elliptic curve groups, the runtime was formidable and required a significant number of bits. In this section, we will discuss some of the advances that were made to improve the computation of the group order.

4.1. Hasse's Theorem. Let E be an elliptic curve over the finite field $\mathbb{F}_q$. Then the order of $E(\mathbb{F}_q)$, denoted by n, satisfies

$$|q + 1 - n| \leq 2\sqrt{q}. \tag{4}$$

This theorem places a bound on the order of a group in a given finite field. Although this theorem does not give an explicit construction of how to find the order of a group, it does provide substantial information regarding the universe of possibilities for n given a finite field $\mathbb{F}_q$ and an elliptic curve E.

5. The Order of a Point

Given a finite field $\mathbb{F}_q$ and a given point P on an elliptic curve, E, we define the order of P to be smallest integer, k, such that $kP = \infty$. Since every subgroup of $E(\mathbb{F}_q)$ can be generated by a point on the elliptic curve, we also see that the value of k must divide the order of $E(\mathbb{F}_q)$, that is, the group formed by the elliptic

curve. At the very least, we see that one way to solve this problem would be to take $P + P + P \ldots$ until the sum only produces P again, since the point at infinity is the identity element. For extremely large groups however, this could prove to be a daunting task. While there is no standard algorithm to finding a solution for k, it is possible through several algorithms to significantly reduce the time needed to solve for such a k, especially compared to a random approach.[3] While one could compute the order of the group using Schoof's Algorithm, at the very least we can rely on Lagrange's Theorem to reduce the set of candidates for k.[4] Then, from here, we can use one of the many factoring algorithms to factor the group order, or one can use one of the algorithms that have been developed to tackle the discrete logarithm problem.[5]

5.1. Baby Step, Giant Step Algorithm. Given a finite field, $\mathbb{F}_q$, and a point, P, on an elliptic curve, E, we can utilize the Baby Step, Giant Step algorithm to find the order of P. By definition, the order of P is defined as the smallest integer, k, such that $kP = \infty$. Suppose the order of the group, $E(\mathbb{F}_q)$, is equal to n. Although we may not know n explicitly, from Hasse's Theorem we know the bounds for n. From (4), we have $q + 1 - 2\sqrt{q} \leq n \leq q + 1 + 2\sqrt{q}$. To check each of the values for n would require checking a total of $4\sqrt{q}$ values.[6] With the Baby Step, Giant Step method, we can reduce the number of values that we need to check substantially. In fact, with this method, we can reduce the number checks from $4\sqrt{q}$ to $4\sqrt[4]{q}$. The following algorithm is taken from **[1]**:

1. Let $Q = (q+1)P$, and $k = -m$.
2. We want to choose an $m \in \mathbb{N}$ such that $m > \sqrt[4]{q}$.
3. Compute and store the points jP for $j = 0, 1, \ldots, m$.
4. Compute the point $Q + k(2mP)$.
5. If $Q + k(2mP) = \pm jP$ for some stored jP, continue to the next step. Otherwise, let $k = k + 1$ and return to Step 4.
6. We can conclude that $(q + 1 + 2mk \mp j)P = \infty$, and we let $M = q + 1 + 2mk \mp j$.
7. Factor M into distinct prime factors $p_1, p_2, \ldots, p_r$, and let $i = 1$.
8. Compute $(M/p_i)P$. If $(M/p_i)P = \infty$, let $M = (M/p_i)$, and return to Step 7. Otherwise continue to the next step.
9. If $i = r$, continue to the next step. Otherwise, let $i = i + 1$, and return to Step 8.
10. M is the order of the point.

[3] We will see later that the inherent difficulty in finding such a k is what makes the elliptic curve-based cryptography system resistant to outside party access. That is, if there were a straight-forward algorithm to cleanly solving the value for k, there would be no reason to use elliptic curve cryptography.

[4] Lagrange's Theorem states that the order of a group element divides the order of the group. This is a particularly useful fact in cases where the group order is extremely large, as one can quickly eliminate a large number of irrelevant candidates for k. As discussed earlier in this paragraph, since the point serves as a generator for some subgroup, Lagrange's Theorem applies both to subgroups and to individual elements.

[5] It is worth noting that to find the order of a point, P, it is not necessary to solve the discrete logarithm problem, as one can find the order of the group and use Hasse's Theorem to narrow the set of candidates substantially, and then use Baby Step, Giant Step algorithm to find the point order.

[6] This is because we would need to check each value of n in the interval $[q + 1 - 2\sqrt{q}, q + 1 + 2\sqrt{q}]$, which makes for a total of $(q + 1 + 2\sqrt{q}) - (q + 1 - 2\sqrt{q}) = 4\sqrt{q}$ values for n.

11. (Optional) To find the order of the group, we can choose random points on E and repeat Steps 1 - 10 until the least common multiple of the orders divides only one integer in the range guaranteed from Hasse's Theorem.[7]

5.2. Point Order Example. Let $\mathbb{F} = \mathbb{F}_7$, and let E be defined by $y^2 = x^3 - x$. Suppose we have the point P on E, defined by $(4, 2)$, and suppose we want to find the order of P. Then, we can do so in one of several ways. Since our group order is likely relatively small, we will demonstrate the case where we continue to add until we arrive back at P. Here we will use Hasse's Theorem from 4.1. To begin, we calculate the bounds for the order of the group. Using (4), we calculate that the bounds for the order of the group are

$$|8 - n| \leq 2\sqrt{7}. \tag{5}$$

Simplifying, we have

$$8 - 2\sqrt{7} \leq n \leq 8 + 2\sqrt{7}. \tag{6}$$

After rounding to whole numbers, we conclude that

$$3 \leq n \leq 13. \tag{7}$$

After utilizing Hasse's Theorem, we can utilize the Baby Step, Giant Step algorithm. This will be faster than simply trying every possible n in the interval.

1. First we compute $Q = (q + 1)/P$. This gives $Q = (8)P$. From 3.3, we can use our scalar multiplication algorithm to eventually calculate that $Q = 8P = (0, 1)$.
2. Next we choose an integer $m > \sqrt[4]{q}$. Since $\sqrt[4]{q} = \sqrt[4]{7} \approx 1.63$, we can choose an $m \geq 2$. Suppose we let $m = 3$.
3. Next we compute the points jP for $j = 0, 1, 2, 3$. Then we have (0,0), (4,2), (1,0), (4,5), respectively.
4. Next we compute $Q + (-m)(2mP)$. Then we have $Q - (2)(9)P = (0, 1) - 18P = (1, 0)$. Since this is equal to jP, for $j = 2$, we can advance to the next step.
5. Then we have $M = 8 + (2)(-3)(3) - j = -10 - 2 = -12$.
6. We factor M as the product of $-1 \times 2 \times 2 \times 3$.
7. We first check $(M/2)P$ and see that $(M/2)P \neq \infty$. Thus, we next check $(M/3)P$, and we see that $(M/3)P = \infty$, thus we can let $M = (M/3)$, giving us $M = -4$.
8. Since the only prime factor for M is 2, we can conclude that the order of P is 4.

We can verify that 4 is the order of P by checking that $4P + P = P$. Checking this condition, from 3.1 we see that $(0, 1) + (4, 2) = (4, 2)$. Hence, we have successfully found the order of the point $(4, 2)$ over the finite field $\mathbb{F}_7$ to be 4.

[7] Using this method, it is clear that it is easier to find the order of an individual point than to find the order of the entire group.

CHAPTER 4

An Introduction to Elliptic Curve Cryptography

1. Introduction

In 1985, Victor Miller and Neal Koblitz independently developed the first elliptic curve based cryptographic encryption system. The advantages of an elliptic curve based cryptographic encryption system were immediately clear: smaller key sizes could generate equivalent levels of information security. Whereas the Diffie-Hellman exchange had already demonstrated the advantages of using group structure for integers, elliptic curve cryptography was the first instance in which algebraic structures (specifically elliptic curves here) that formed a group, were utilized.[1] With the development of elliptic curve cryptography came a newfound curiosity among the mathematical community. After witnessing the marriage of analytic number theory and analytic geometry with cryptography, many mathematicians shifted their focus towards applying their mathematical expertise to cryptographic encryption systems. One result of this trend was the homomorphic encryption system, which is briefly discussed later in this section. To better understand how the encryption and decryption process works in an elliptic curve based system, we will outline an example of a signature scheme later in this chapter, which is often used to validate the identity of the sender of a message.

2. A Digital Signature Example in ECC

Suppose person A wishes to communicate the message 3 with person B. As with any public key system, we will have a total of four keys: public key A, private key A, public key B, and private key B. For a given finite field $\mathbb{F}_p$, from Galois Theory, since $\mathbb{F}_p$ is finite, we have a subgroup, H, with prime order n and a point generator P. We see that P is a point on the elliptic curve since P^n is the same as P added to itself n times, which is equal to the point at infinity based on our definition of the point order. In order to construct both the public key and private key, person A will need to randomly choose an integer in [1, n - 1]. Note that this initial process is very similar to the RSA key construction process, with the exception that in this case we need only utilize one random number.

Suppose person A chooses the random integer x in [1, n - 1]. Then, we have private key A $= x$, and public key A $= Q := xP$. Note that in elliptic curve cryptography, the public key is a point on the elliptic curve, whereas the private key is an integer. Suppose $Q = (a, b)$. Then we next compute $r = a \bmod n$. If r is an element of (0, n - 1], we can advance to the next step; however, if $r = 0$, we must go back to our initial step and choose a new value for x. Suppose we have achieved r not equal to 0. Then we compute $x^{-1} \bmod n$, and let $s = x^{-1}(3 + xr)$

[1]The Diffie-Hellman system utilizes group structure formed from integer multiplication in modulo p, for a prime p.

mod n. If s is an element of (0, n - 1], we can advance to the next step; however, if $s = 0$, we must go back to our initial step and choose a new value for x.

Suppose we have achieved s not equal to 0. Then we write the message signature for person A as (r, s). For person B to verify that it was in fact person A who sent the message, he/she will need to introduce a similar method. At person B's disposal is the value Q (public key A), the value of n, the value of point generator P, as well as the signature (r, s). Using this information, person B can calculate the numerical value of the message, in this case 3. Next, person B calculates $y = 3s^{-1}$ mod n and $z = rs^{-1}$ mod n. From here, person B computes $(f, g) := yP + zQ$. Lastly, person B computes $v := f$ mod n, with v nonnegative. If $v = r$, person B has successfully verified the identity of the sender.

3. Advantages of Elliptic Curve Cryptography

One of the most significant advantages of elliptic curve cryptography is its ability to generate similar levels of security with a much smaller key than, for example, an RSA construction. This ability to utilize smaller keys signifies a computational advantage and time advantage when compared to other asymmetric key systems. Additionally, as an asymmetric key system, it is generally safer than a symmetric key system in that it is more difficult to decrypt without the private key.

4. Disadvantages of Elliptic Curve Cryptography

While elliptic curve cryptography is generally safer in terms of its difficulty to break the cipher, it can also generally be very demanding computationally. This increased degree of computation translates into a less time efficient method of encryption, generally requiring more time than other encryption methods to run. Although elliptic curve systems generally run faster than RSA or Diffie-Hellman systems, they are still significantly slower than symmetric key systems, which is why many systems only use an asymmetric key system to distribute keys before eventually utilizing a faster symmetric key system.

Additionally, currently methods exist now such as Pollard's rho method and the baby-step, giant-step method that have reduced the discrete logarithm problem in elliptic curves to a more manageable computational problem, albeit the problem is only reduced to one with run time $O(\sqrt{n})$ where n is the order of the group. Although solving the discrete logarithm is still unrealistic in short time periods through these methods, many argue that quantum computers would be able to drastically reduce the security of the cryptographic system using either of the two mentioned methods. For this reason, efforts are currently being made to develop new cryptographic systems that will be better suited for a quantum-based landscape. As we will discuss in Chapter 7, the NSA has already made a statement advising against the further development of elliptic curve cryptography going forward.

5. Applications of Elliptic Curve Cryptography

The main applications of elliptic curve cryptography include online banking, currency transfers, commerce, and digital signatures. As an example, Bitcoin is a digital currency that is peer-to-peer distributed, in contrast to wire transfers or other forms of financial transfers that utilize a bank or other intermediary. For this

reason, elliptic curve cryptography is an important tool for securing the transfer of Bitcoin. Beyond this, the most popular use for elliptic curve cryptography is in fact authentication of the identity of the sender, making the digital signature application the most utilized feature.

CHAPTER 5

Data

1. SageMath

1.1. Introduction. For this research, we relied heavily on SageMath, an open source mathematical software, to compute the point orders that are recorded in the tables within the sections that follow. SageMath operates similarly to Python, and has built-in functionality to work with elliptic curves and finite fields.

1.2. Example Code. The following demonstrates code that can be used to generate the average point order for the point $(x, y) = (4, 3)$ in elliptic curves that contain (4,3) in a finite field with order 47.

```
t = 47
F.<g> = GF(t)
p = (4,3)
m = p[0]
n = p[1]
l = [ ]
for a in xrange(0,t):
    b = pow(n,2) - pow(m,3) - (a*m)
    if (4*pow(a,3) + 27*pow(b,2)) % t != 0:
        E = EllipticCurve(F, [0,0,0,a,b])
        l.append(E(p).order())
d = float(sum(l))/float(len(l))
print d
```

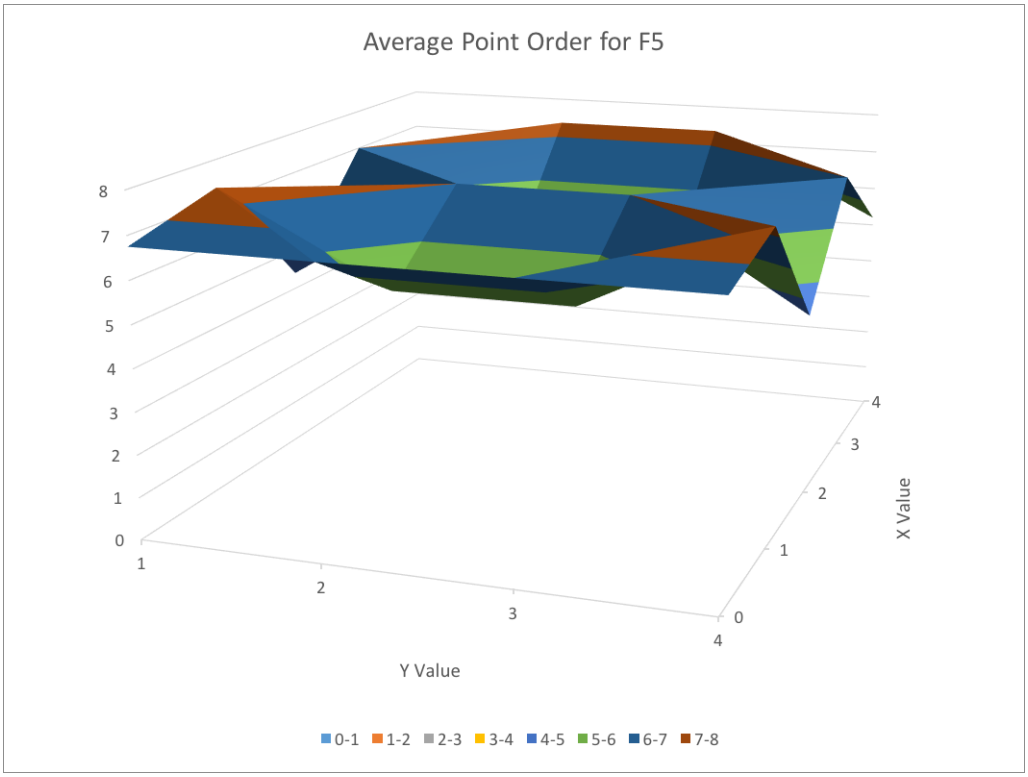
The value that is returned, d, is 26.1914893617.

2. Field of Order 5

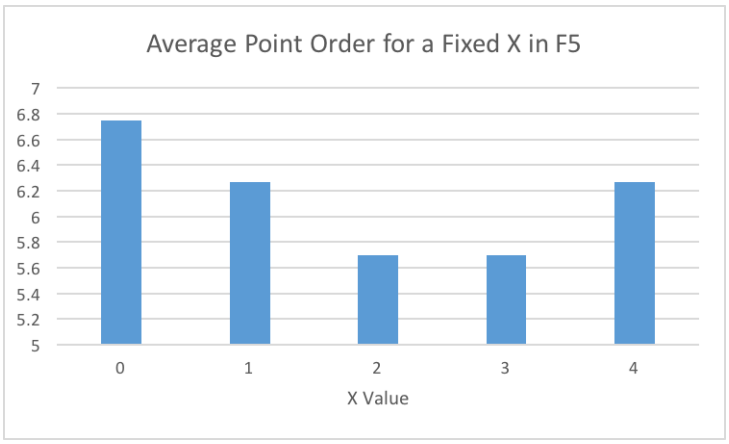
2.1. Table of Average Point Orders.

x\y	1	2	3	4
0	6.75	6.75	6.75	6.75
1	7.33333333	5.2	5.2	7.33333333
2	4.4	7	7	4.4
3	7	4.4	4.4	7
4	5.2	7.3333333	7.3333333	5.2

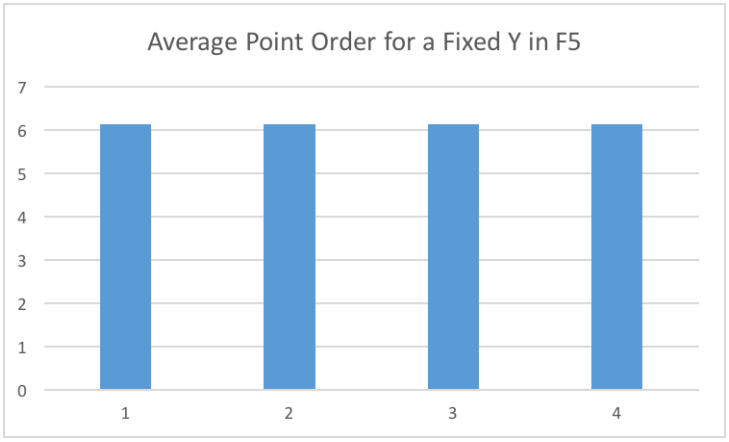
2.2. Graph of Average Point Orders.



2.3. Average of Average Point Orders for each Value of X.



2.4. Average of Average Point Orders for each Value of Y.

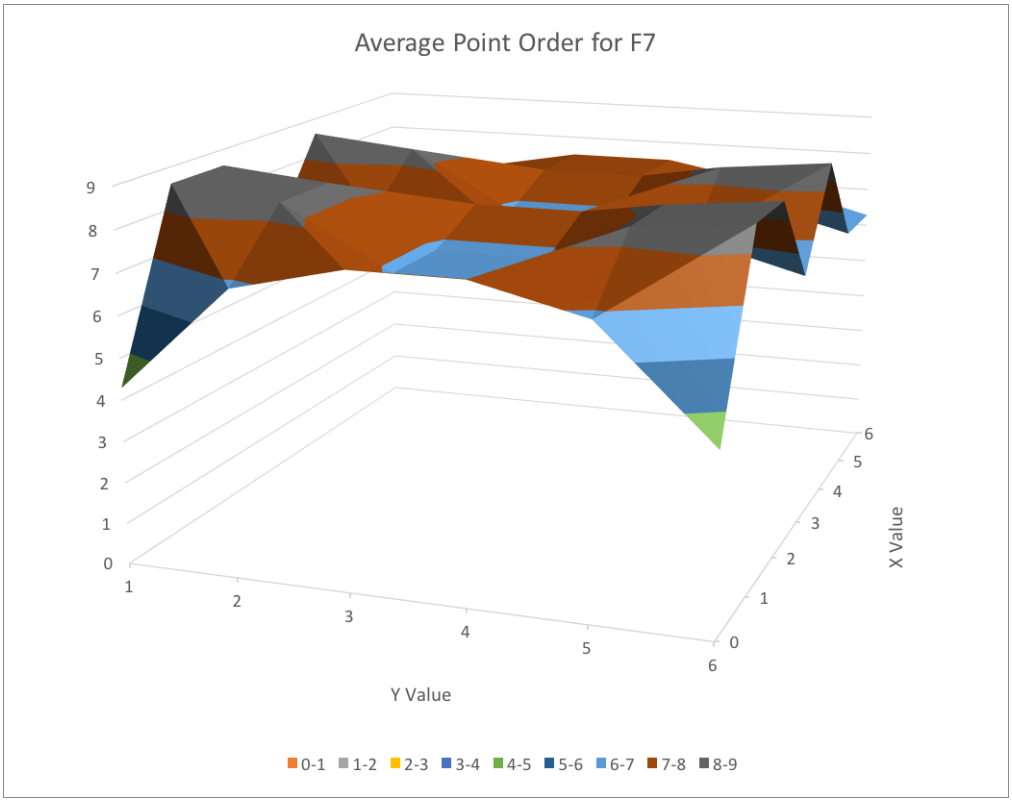


3. Field of Order 7

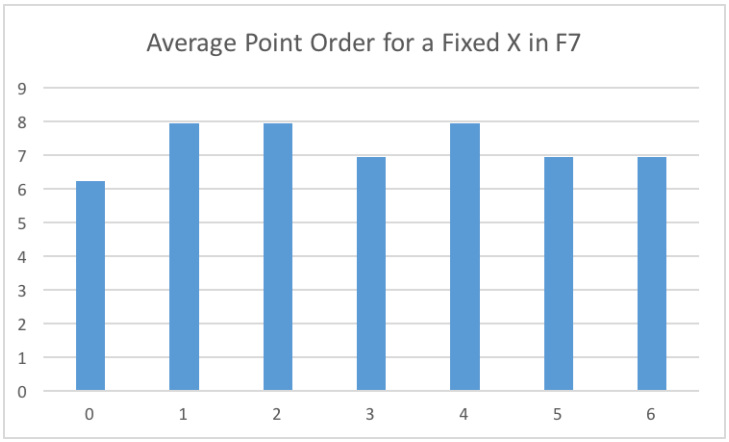
3.1. Table of Average Point Orders.

x\y	1	2	3	4	5	6
0	4.2857	6.8571	7.5	7.5	6.8571	4.2857
1	8.6	8.3333	6.8571	6.8571	8.3333	8.6
2	8.6	8.3333	6.8571	6.8571	8.3333	8.6
3	6.2857	7	7.5	7.5	7	6.2857
4	8.6	8.3333	6.8571	6.8571	8.3333	8.6
5	6.2857	7	7.5	7.5	7	6.2857
6	6.2857	7	7.5	7.5	7	6.2857

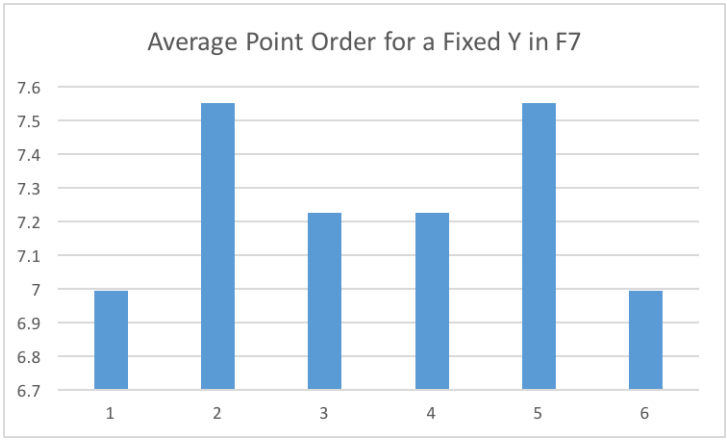
3.2. Graph of Average Point Orders.



3.3. Average of Average Point Orders for each Value of X.



3.4. Average of Average Point Orders for each Value of Y.

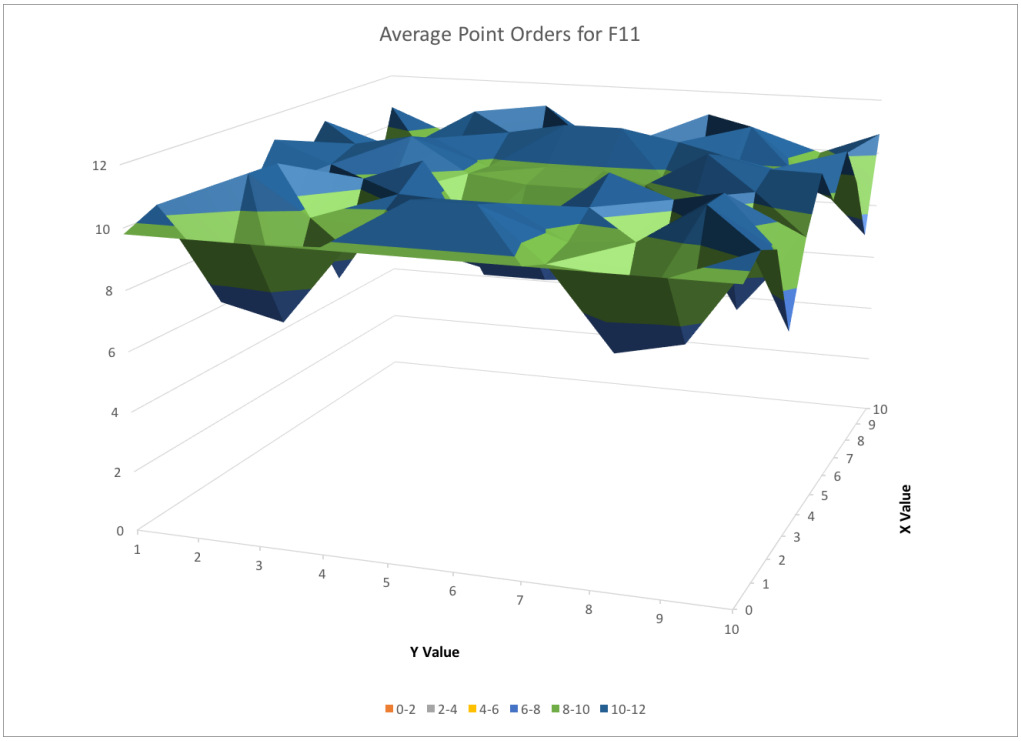


4. Field of Order 11

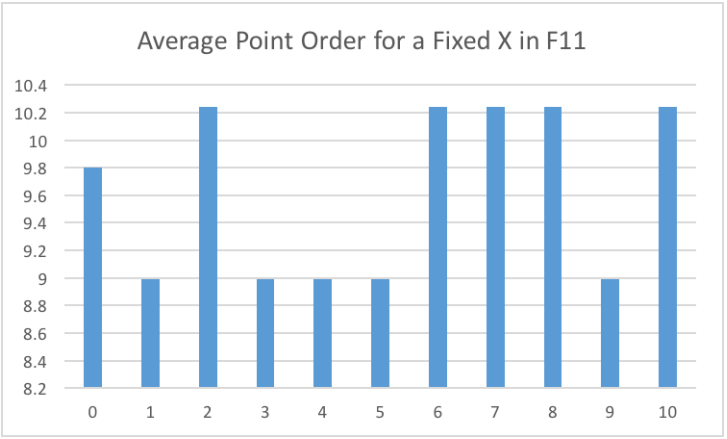
4.1. Table of Average Point Orders.

x\y	1	2	3	4	5	6	7	8	9	10
0	9.8	9.8	9.8	9.8	9.8	9.8	9.8	9.8	9.8	9.8
1	10.3333	7.3	6.8182	9.4	11.0909	11.0909	9.4	6.8182	7.3	10.3333
2	9.9	11.125	9.8	9.6364	10.7273	10.7273	9.6364	9.8	11.125	9.9
3	6.8182	11.0909	7.3	10.3333	9.4	9.4	10.3333	7.3	11.0909	6.8182
4	9.4	6.8182	10.3333	11.0909	7.3	7.3	11.0909	10.3333	6.8182	9.4
5	11.0909	10.3333	9.4	7.3	6.8182	6.8182	7.3	9.4	10.3333	11.0909
6	9.8	10.7273	11.125	9.9	9.6364	9.6364	9.9	11.125	10.7273	9.8
7	11.125	9.6364	10.7273	9.8	9.9	9.9	9.8	10.7273	9.6364	11.125
8	9.6364	9.8	9.9	10.7273	11.125	11.125	10.7273	9.9	9.8	9.6364
9	7.3	9.4	11.0909	6.8182	10.3333	10.3333	6.8182	11.0909	9.4	7.3
10	10.7273	9.9	9.6364	11.125	9.8	9.8	11.125	9.6364	9.9	10.7273

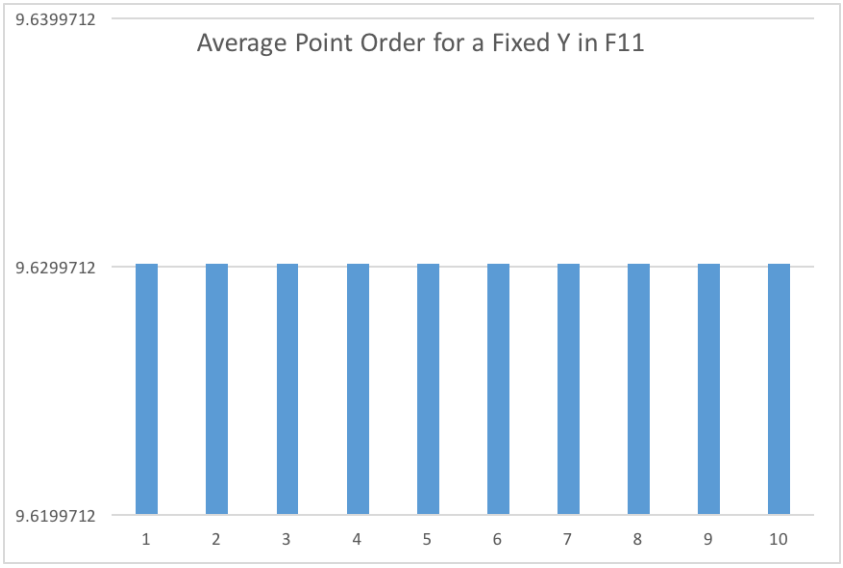
4.2. Graph of Average Point Orders.



4.3. Average of Average Point Orders for each Value of X.



4.4. Average of Average Point Orders for each Value of Y.

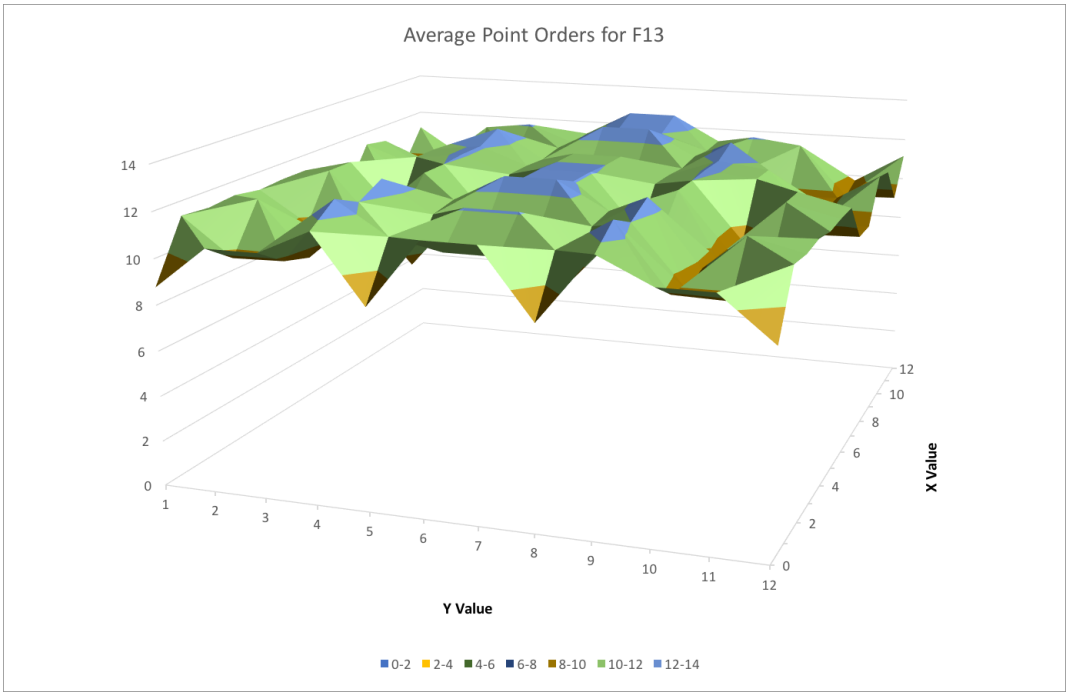


5. Field of Order 13

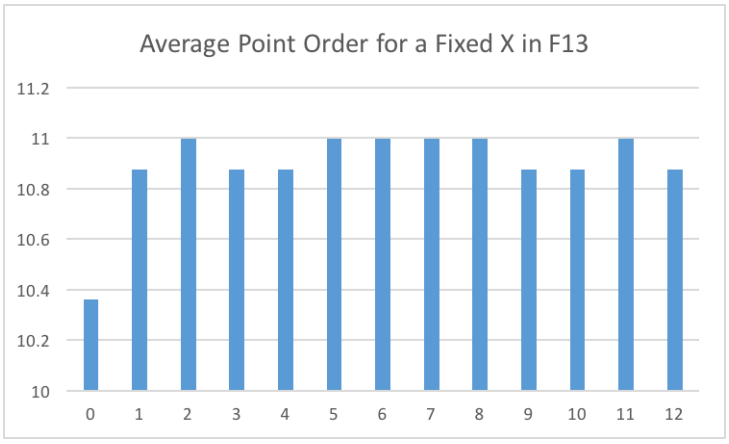
5.1. Table of Average Point Orders.

x\y	1	2	3	4	5	6	7	8	9	10	11	12
0	8.7692	10.62	10.62	11.7	8.769	11.7	11.7	8.769	11.7	10.62	10.62	8.769
1	11.364	9.692	9.75	12.58	11.17	10.69	10.692	11.17	12.583	9.75	9.692	11.36
2	11.25	11.8	9.385	12	9.462	12.08	12.083	9.462	12	9.385	11.8	11.25
3	11.364	9.692	9.75	12.58	11.17	10.69	10.692	11.17	12.583	9.75	9.692	11.36
4	11.167	9.75	9.692	10.69	11.36	12.58	12.583	11.36	10.692	9.692	9.75	11.17
5	11.25	11.8	9.385	12	9.462	12.08	12.083	9.462	12	9.385	11.8	11.25
6	11.25	11.8	9.385	12	9.462	12.08	12.083	9.462	12	9.385	11.8	11.25
7	9.4615	9.385	11.8	12.08	11.25	12	12	11.25	12.083	11.8	9.385	9.462
8	9.4615	9.385	11.8	12.08	11.25	12	12	11.25	12.083	11.8	9.385	9.462
9	11.364	9.692	9.75	12.58	11.17	10.69	10.692	11.17	12.583	9.75	9.692	11.36
10	11.167	9.75	9.692	10.69	11.36	12.58	12.583	11.36	10.692	9.692	9.75	11.17
11	9.4615	9.385	11.8	12.08	11.25	12	12	11.25	12.083	11.8	9.385	9.462
12	11.167	9.75	9.692	10.69	11.36	12.58	12.583	11.36	10.692	9.692	9.75	11.17

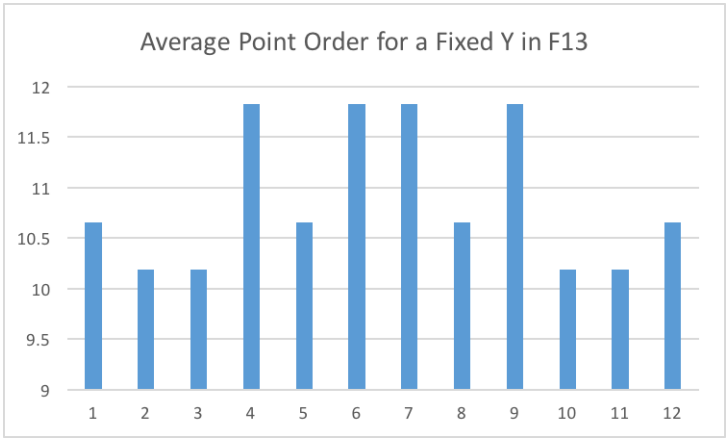
5.2. Graph of Average Point Orders.



5.3. Average of Average Point Orders for each Value of X.



5.4. Average of Average Point Orders for each Value of Y.

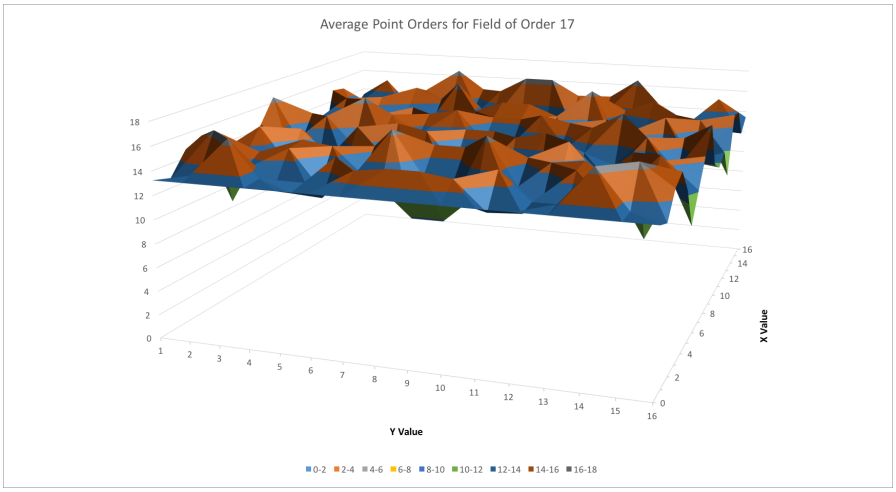


6. Field of Order 17

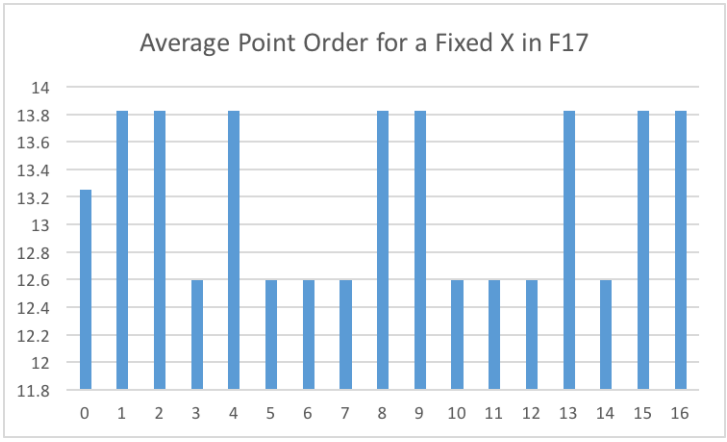
6.1. Table of Average Point Orders.

x\y	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	13.25	13.25	13.25	13.25	13.25	13.25	13.25	13.25	13.25	13.25	13.25	13.25	13.25	13.25	13.25	13.25
1	12.73	13.35	16.50	13.44	12.47	12.31	14.71	15.07	15.07	14.71	12.31	12.47	13.44	16.50	13.35	12.73
2	14.71	16.50	13.44	12.31	12.73	15.07	13.35	12.47	12.47	13.35	15.07	12.73	12.31	13.44	16.50	14.71
3	15.38	9.88	14.38	14.86	11.75	12.56	12.12	9.82	9.82	12.12	12.56	11.75	14.86	14.38	9.88	15.38
4	13.35	13.44	12.31	15.07	14.71	12.47	16.50	12.73	12.73	16.50	12.47	14.71	15.07	12.31	13.44	13.35
5	9.82	15.38	12.12	9.88	12.56	14.38	11.75	14.86	14.86	11.75	14.38	12.56	9.88	12.12	15.38	9.82
6	12.12	14.38	14.86	12.56	15.38	9.82	9.88	11.75	11.75	9.88	9.82	15.38	12.56	14.86	14.38	12.12
7	14.38	12.56	9.82	11.75	9.88	15.38	14.86	12.12	12.12	14.86	15.38	9.88	11.75	9.82	12.56	14.38
8	16.50	12.31	15.07	12.47	13.35	12.73	13.44	14.71	14.71	13.44	12.73	13.35	12.47	15.07	12.31	16.50
9	12.47	14.71	13.35	16.50	15.07	13.44	12.73	12.31	12.31	12.73	13.44	15.07	16.50	13.35	14.71	12.47
10	11.75	12.12	9.88	14.38	9.82	14.86	15.38	12.56	12.56	15.38	14.86	9.82	14.38	9.88	12.12	11.75
11	12.56	11.75	15.38	12.12	14.86	9.88	9.82	14.38	14.38	9.82	9.88	14.86	12.12	15.38	11.75	12.56
12	9.88	14.86	12.56	9.82	12.12	11.75	14.38	15.38	15.38	14.38	11.75	12.12	9.82	12.56	14.86	9.88
13	15.07	12.73	14.71	13.35	12.31	16.50	12.47	13.44	13.44	12.47	16.50	12.31	13.35	14.71	12.73	15.07
14	14.86	9.82	11.75	15.38	14.38	12.12	12.56	9.88	9.88	12.56	12.12	14.38	15.38	11.75	9.82	14.86
15	12.31	12.47	12.73	14.71	13.44	13.35	15.07	16.50	16.50	15.07	13.35	13.44	14.71	12.73	12.47	12.31
16	13.44	15.07	12.47	12.73	16.50	14.71	12.31	13.35	13.35	12.31	14.71	16.50	12.73	12.47	15.07	13.44

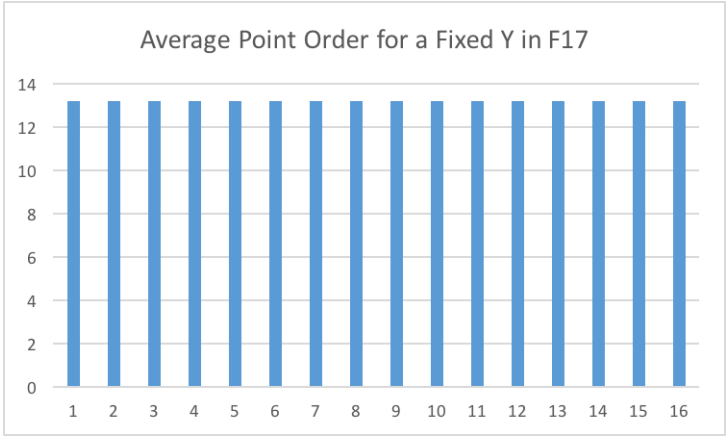
6.2. Graph of Average Point Orders.



6.3. Average of Average Point Orders for each Value of X.



6.4. Average of Average Point Orders for each Value of Y.



CHAPTER 6

Analysis

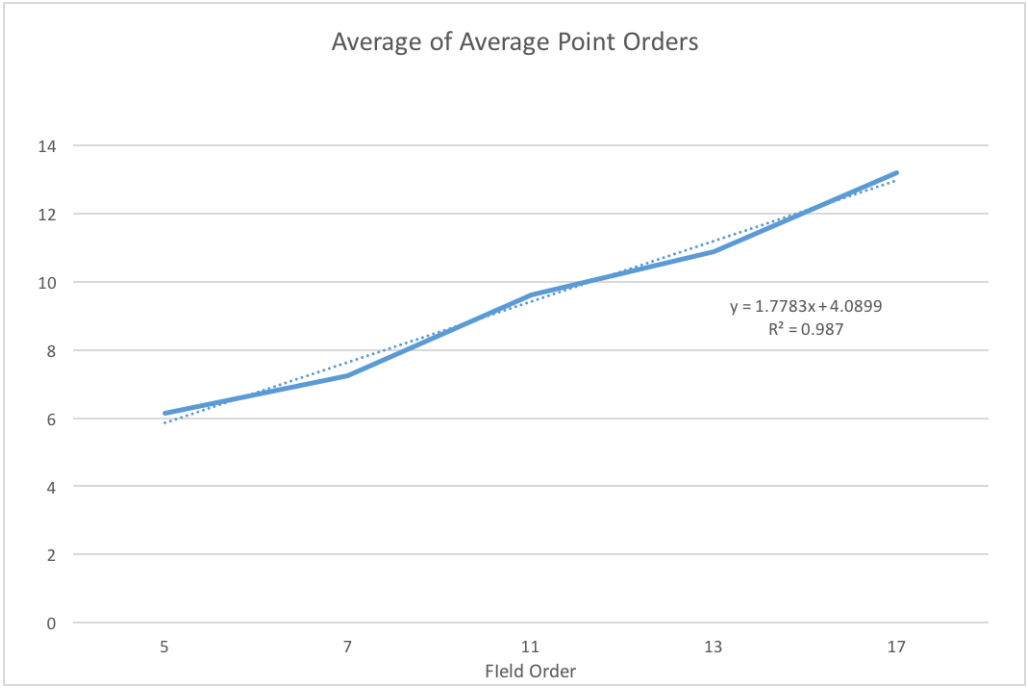
1. Group Structure

In working with a group structure under elliptic curves, one would certainly expect symmetry in element orders. Our data reflects this symmetric nature, through our average point orders for a given field. This consistency with the group structure is especially striking from the graphs of the average point orders when y is fixed, where the graphs are symmetric about $y = p/2$ for $\mathbb{F}_p$.

2. Mean Calculation Method

One aspect of our data that is worth pointing out is the method by which the average was calculated in each of the tables and graphical models. While for a given point, the true point average is represented, for the sake of ease in calculation, averages listed for the entire field and for a fixed x or y value were calculated using an average of averages approach. While this value is still a good approximation for the true average, in truth this method has the potential to slightly distort the true average for a given field or for a fixed x or y value within a field.

3. Average Point Orders in Relation to Size of Group



Based on our results, we would conjecture that the order goes to infinity as the size of the field goes to infinity. However, based on our relatively small sample size of fields used and the relatively small size of fields used, that conclusion is certainly an extrapolation. Moreover, while we have included linear regression analysis in analyzing our data, a larger trend may exist that does not generally conform to a linear model, making the linear model potentially inadequate in predicting average point orders for a given field.

4. Future Research

One natural question that emerges from this research is what the probability distribution looks like in obtaining a certain point order for a given field. A barrier to producing such a distribution from our results is the fact that we are working with the average order for a given point, as measured by the mean of the order that the point has in each of the elliptic curves which pass through that point for the given field. As discussed in Section 2, using our data to produce such a distribution would distort the actual distribution, as some averages would hold different weightings than if measured on an individual point order basis. In that sense, future research could examine, for a fixed field, the averages on an individual basis, rather than an average of averages approach.

CHAPTER 7

The Future of Cryptography

1. Trends in Development

As more efficient computers continue to develop, larger keys will be required to be put into practice. Thus, the trend for larger keys and faster computing will continue to drive cryptography towards more complex structures and more difficult computing.

2. The Future of Elliptic Curve Cryptography

In August of 2015, the NSA released a statement advising against the development of elliptic curve cryptographic encryption systems, as the organization recognized that the rise of quantum computers will largely reduce the difficulty of efficiently solving the discrete logarithm problem, thereby reducing the time for unwanted third parties from gaining access to an encrypted message. An excerpt from the NSA statement reads as follows:

> For those partners and vendors that have not yet made the transition to Suite B algorithms, we recommend not making a significant expenditure to so at this point but instead to prepare for the upcoming quantum resistant algorithm transition... Unfortunately, the growth of elliptic curve use has bumped up against the fact of ontinued progress in the research on quantum computing, necessitating a re-evaluation of our cryptographic scheme. **[7]**

Though it is unclear what progress has been made, what is clear is that the NSA has begun efforts to develop such a quantum computer. As of 2014, the NSA's *Penetrating Hard Targets* program was allocated $79.9 million towards the development of a quantum computer.**[10]** Despite this, many computer scientists and experts in the field argue that it is extremely unlikely that the NSA has already successfully constructed such a working computer. Moreover, most experts believe that the NSA is likely in step with current research surrounding quantum computers. For this reason, many acknowledge the statement of the NSA as reflective of the general progress that has been made in quantum computing, as opposed to some mysterious hint that the NSA may have successfully implemented such a technology.

Although quantum computers will certainly change the face of cryptography, the landscape will remain, as new encryption methods will follow to combat the rise of a more powerful machine. More than likely, new encryption schemes that have yet to be created will be possible only through the development of such a quantum computer.

3. Homomorphic Cryptography

More recently, a new cipher development in the public key system, called homomorphic encryption, is becoming increasingly popular in securing communications. In 2009, Craig Gentry published a thesis outlining the first ever fully homomorphic encryption scheme, which utilized ideal lattices. Since then, more advanced methods, that do not require ideal lattices, have been presented by others that are more efficient for practical use. While a majority of businesses and government entities still use elliptic curve cryptography in their combination of public and private key communication measures, the number of homomorphic encryption users is growing.

4. Quantum Cryptography

Theorized in the 1970's, Quantum cryptography relies on the use of quantum key distribution as a means of allowing for secure key exchange and consequently to a more secure information exchange. Although it has yet to be implemented, quantum key distribution would hold a significant advantage in that it would allow for communicating parties to know whether or not a breach has occurred. Beyond simply making it more difficult for unwanted third parties to obtain access to information, this system would utilize quantum properties that would actually detect an eavesdropper, something which is not possible with current cryptographic systems, thereby making this theorized system extremely attractive. Moreover, this feature would largely eliminate the need for a digital signature process. As quantum computing continues to improve, this type of cryptographic scheme is becoming more and more realistic, and according to many experts, such a system could surface within the next ten to fifteen years.

CHAPTER 8

Acknowledgements

I would first and foremost like to thank Dr. Qiao Zhang for his unwavering support and direction throughout the research process. As a former student of his in algebra, I was privileged enough to continue working with him in the form of this thesis. He has been an exceptional mentor to me personally, and without him the completion of this project would have certainly been impossible. His knowledge and insights have been a tremendous resource, and it was an honor to be able to work with him throughout the last year.

In addition, I would like to thank Dr. Ze-Li Dou and Dr. Liran Ma for their help in editing this document as well as providing input that proved helpful in completing the research.

Bibliography

[1] Washington, Lawrence C.Elliptic Curves: Number Theory and Cryptography. Boca Raton: Chapman and Hall, 2008. Print.

[2] Silverman, Joseph H.The Arithmetic of Elliptic Curves. New York: Springer-Verlag, 1986. Print.

[3] Bard, Gregory V.Sage for Undergraduates. Providence: AMS, 2015. Print.

[4] "The Black Chamber." - Pigpen Cipher. Simon Singh. Web. 02 Apr. 2016.

[5] Menezes, A. J., Van Oorschot Paul C., and Scott A. Vanstone.Handbook of Applied Cryptography. Boca Raton: CRC, 1997. Print.

[6] Koblitz, Ann Hibner, Neal Koblitz, and Alfred Menezes. "Elliptic Curve Cryptography: The Serpentine Course of a Paradigm Shift."Publications. International Association for Cryptologic Research, 28 Aug. 2008. Web. https://eprint.iacr.org/2008/390.pdf.

[7] Koblitz, Neal, and Alfred J. Menezes. "A Riddle Wrapped in an Enigma." IACR (2015). Web.

[8] Markoff, John. "Cryptography Pioneers Win Turing Award." The New York Times 1 Mar. 2016. Web.

[9] Apuzzo, Matt. "F.B.I. Used Hacking Software Decade Before IPhone Fight." The New York Times 13 Apr. 2016, Technology sec. Web.

[10] Rich, Steven, and Barton Gellman. "NSA Seeks to Build Quantum Computer That Could Crack Most Types of Encryption." The Washington Post 2 Jan. 2014, National Security sec. Web.

[11] Bos, Joppe W., J. Alex Halderman, Nadia Heninger, Jonathan Moore, Michael Naehrig, and Eric Wustrow. "Elliptic Curve Cryptography in Practice." Financial Cryptography and Data Security Lecture Notes in Computer Science (2014): 157-75. Web.