

## Union College Union | Digital Works

---

Honors Theses

Student Work

---


6-2017

# Elliptic Curve Cryptology

Francis Rocco

*Union College - Schenectady, NY*

Follow this and additional works at: <https://digitalworks.union.edu/theses>

 Part of the [Information Security Commons](#), [Mathematics Commons](#), and the [Special Functions Commons](#)

---

### Recommended Citation

Rocco, Francis, "Elliptic Curve Cryptology" (2017). *Honors Theses*. 74.  
<https://digitalworks.union.edu/theses/74>

This Open Access is brought to you for free and open access by the Student Work at Union | Digital Works. It has been accepted for inclusion in Honors Theses by an authorized administrator of Union | Digital Works. For more information, please contact [digitalworks@union.edu](mailto:digitalworks@union.edu).

# Elliptic Curve Cryptology

By

**Francis Rocco**

\* \* \* \* \*

Submitted in partial fulfillment of the requirements for  
Honors in the Department of Mathematics

Union College

March, 2017

## ABSTRACT

ROCCO, FRANCIS    Elliptic Curve Cryptology.

Department of Mathematics, March 2017.

ADVISOR: HATLEY, JEFFREY

In today's digital age of conducting large portions of daily life over the Internet, privacy in communication is challenged extremely frequently and confidential information has become a valuable commodity. Even with the use of commonly employed encryption practices, private information is often revealed to attackers. This issue motivates the discussion of cryptology, the study of confidential transmissions over insecure channels, which is divided into two branches of cryptography and cryptanalysis. In this paper, we will first develop a foundation to understand cryptography and send confidential transmissions among mutual parties. Next, we will provide an expository analysis of elliptic curves and then utilize them to strengthen our cryptographic methods. Finally, we will discuss cryptanalytic attacks against our confidential transmissions and ultimately detail how to best choose elliptic curves that are cryptographically robust.

## ACKNOWLEDGEMENT

I am immensely grateful for my advisor, Professor Jeffrey Hatley, without whom this thesis would not have been possible. His precise attention to detail was unrivaled, his endless patience for my countless questions was greatly appreciated, and he ensured that this thesis became exactly what I wanted it to be.

## NOTATION

We shall use the following notation throughout this paper. We write  $\mathbf{N}$  for the set of natural numbers,  $\mathbf{Z}$  for the set of integers,  $\mathbf{Q}$  for the set of rational numbers,  $\mathbf{R}$  for the set of real numbers,  $p$  will always denote a prime, and  $\mathbf{Z}/p\mathbf{Z}$  for the set of integers modulo  $p$ .  $\mathbf{F}_p \cong \mathbf{Z}/p\mathbf{Z}$  is a finite field with  $p$  elements, and  $\mathbf{F}_p^\times = \mathbf{F}_p \setminus \{0\}$  is the multiplicative group of units with  $p - 1$  elements.

## CONTENTS

ABSTRACT	ii
ACKNOWLEDGEMENT	iii
NOTATION	iv
1. INTRODUCTION	1
1.1. An Introduction to Cryptography	1
2. KEY EXCHANGES	7
2.1. The Discrete Logarithm Problem	8
2.2. Diffie-Hellman Key Exchange	10
2.3. The Index Calculus Algorithm	12
3. ELLIPTIC CURVES	16
3.1. Introduction	16
3.2. Elliptic Curves and a Group Law	17
3.3. DHP and Elliptic Curves	30
4. CRYPTANALYSIS: ATTACKS AGAINST THE ECDLP	31
4.1. The Baby-Step, Giant-Step Algorithm	32
4.2. Pollard's $\rho$ Algorithm	34
4.3. Cryptographically Robust Elliptic Curves	37
APPENDIX: APPLICATIONS AND IMPLEMENTATION	38
References	40

## 1. INTRODUCTION

1.1. **An Introduction to Cryptography.** In recent times, cybersecurity has become an increasingly popular issue among the general public. With the exponential advancements of technology and the abundance of personal data circulating the Internet, it is significantly more difficult to conceal personal information from prying eyes. Though security measures have grown more complex, they are hindered by ongoing attacks aimed at stealing information. How can we approach the issue of a proper level of security without compromising either the efficiency of transmitting messages or the difficulty in intercepting them, especially when there are new methods and efforts being developed regularly? The problem presented motivates a deeper understanding of *cryptography* - the study of transmitting confidential information over unprotected channels - and how it can be implemented to restore a reasonable level of security for all of the information.

In order to work through examples in cryptography, we will become familiar with the typical nomenclature for hypothetical scenarios. First, we consider two parties that wish to communicate and name them Alice and Bob; they are a great distance from one another, and therefore, their only method of communication is over an insecure channel. Alice wishes to transmit a secret message to Bob without compromising the contents of her message to any other third party. Alice begins with her message, the *plaintext*  $M$ , and then uses an encryption key  $e$  to convert it into a *ciphertext*  $C$ . Ultimately Bob must be able to convert  $C$  back to  $M$  with a *decryption key*  $d$ ; however, in order to preserve secrecy, he should be the only one with that capability. Thus,

we have described the components of a *cryptosystem* involving  $M$ ,  $e$ ,  $C$ , and  $d$ . This system represents a method wherein the message can be sent with only Alice and Bob able to read it. The following is an example to illustrate how such a system operates:

*Example 1. The Caesar Cipher:* This cipher is perhaps the simplest cipher that one could apply to transmit a secret message. We begin by assigning numeric values to letters of the alphabet (namely,  $a = 1$ ,  $b = 2$ ,  $c = 3$ ,  $\dots$ ,  $z = 26$ ). Now, consider an example plaintext  $M$ , “helloworld”, and convert it to a string of numerical values, obtaining the plaintext

$$M = 8 - 5 - 12 - 12 - 15 - 2 - 15 - 2 - 8 - 15 - 23 - 1 - 18 - 5 - 25 - 15 - 21.$$

We add dashes here solely to clarify different letters from one another. For this example, we define  $e = 4$  and add  $e$  to the current value of each letter, yielding

$$12 - 9 - 16 - 16 - 19 - 6 - 19 - 6 - 16 - 19 - 27 - 5 - 22 - 9 - 29 - 19 - 25.$$

Since some of the values resulted in numbers greater than 26, we simply reduce modulo 27 and treat 0 as a \*. Thus, our final ciphertext is

$$C = 12 - 9 - 16 - 16 - 19 - 6 - 19 - 6 - 16 - 19 - 0 - 5 - 22 - 9 - 3 - 19 - 25.$$

Observe that if we converted the numbers back to letters, then it would read “lppsfps\*evicsy”. Certainly this new message does not resemble English, rendering the true meaning unknown to anyone else.



Now, we'll convert  $C$  back into  $M$  using the decryption key  $d$ . Since we initially added one to each letter's numeric value, we will use the decryption key  $d = -4$ , namely, the additive inverse of  $e$ , and reduce modulo 27 if we receive negative numbers. Note here that by knowing  $e$  we were able to determine  $d$  immediately, thus rendering the keys *linked*. Now, we subtract four from each of the values, giving us

$$8 - 5 - 12 - 12 - 15 - 2 - 15 - 2 - 8 - 15 - 23 - 1 - 18 - 5 - 25 - 15 - 21,$$

which is our original  $M$ , as desired.

We now introduce a third party named Eve — she is looking to intercept the message that Alice is transmitting to Bob. Though we just constructed a way for Alice and Bob to transmit secret messages, there is an inherent problem in that there are only 26 possible choices for keys. Eve could easily attempt all 26 scenarios and ultimately decrypt the secret message. We now define the type of cryptosystem used in our example.

**Definition 1.** A **symmetric key cryptosystem** is one in which the encryption key and decryption key are linked in the sense that as soon as either  $e$  or  $d$  is revealed, the other is easily determined.

In contrast, there are also asymmetric key cryptosystems wherein  $e$  and  $d$  are unlinked (entirely separate), but there is a large trade-off between speed and security. A great benefit of symmetric key cryptosystems is that they are much faster and ultimately more efficient to encrypt and decrypt messages when the keys are linked; however, since knowing one immediately reveals the other, they are not necessarily the most secure. Still, there are ways to use symmetric key

cryptosystems and maintain an appropriate level of security. Thus, for the remainder of this paper, we will focus on symmetric key cryptosystems and use the following principle as a guide:

**Kerckhoff's Principle (1888).** *The security of a cryptosystem must not depend on the secrecy of the algorithms, but should rest entirely on the strength of the keys.*

Applying this principle to the cryptosystem that we constructed yields the following: we assume that Eve knows the properties of the cryptosystem being used and has the same computing power as Alice and Bob — however, we want to ensure that it is extremely difficult or infeasible for her to figure out  $M$  from  $C$ . We will now discuss another type of symmetric key cryptosystem that has stronger overall security (that is, keys that are significantly more difficult to determine).

*Example 2. Matrix Multiplication:* Bob wishes to transmit a message to Alice. Consider the plaintext “helloaliceiamgreat”. Using the same method as Example 1, we convert this string into numeric values,

8 – 5 – 12 – 12 – 15 – 1 – 12 – 9 – 3 – 5 – 9 – 1 – 13 – 7 – 18 – 5 – 1 – 20.

Now, we break the string into substrings of  $n$  letters (in this case, we set  $n = 3$ ) and then assign each substring to column vectors  $M_1, M_2, \dots, M_k$ . If  $n$  does not divide the length of the string, then insert 0 entries to complete the last matrix. For this example, this means that our string converts into:

$$M = \begin{bmatrix} 8 \\ 5 \\ 12 \end{bmatrix} \begin{bmatrix} 12 \\ 15 \\ 1 \end{bmatrix} \begin{bmatrix} 12 \\ 9 \\ 3 \end{bmatrix} \begin{bmatrix} 5 \\ 9 \\ 1 \end{bmatrix} \begin{bmatrix} 13 \\ 7 \\ 18 \end{bmatrix} \begin{bmatrix} 5 \\ 1 \\ 20 \end{bmatrix} .$$

Note that the matrices are simply juxtaposed next to one another, since clearly they cannot be multiplied together. Recall that  $\text{GL}_n(\mathbf{Z}/27\mathbf{Z})$  is the group of  $n \times n$  invertible matrices with entries from  $\mathbf{Z}/27\mathbf{Z}$ . Then we choose a matrix,

$$e = A \in \text{GL}_n(\mathbf{Z}/27\mathbf{Z}).$$

Since  $e$  is in  $\text{GL}_n(\mathbf{Z}/27\mathbf{Z})$ , we know that it has an inverse matrix

$$d = A^{-1} \in \text{GL}_n(\mathbf{Z}/27\mathbf{Z}).$$

For this example, we choose

$$e = \begin{bmatrix} 1 & 0 & 2 \\ 1 & 3 & 0 \\ 2 & 2 & 1 \end{bmatrix} .$$

Now, we calculate

$$C = AM := AM_1 \ AM_2 \ \dots \ AM_k \pmod{27}.$$

The juxtaposition of the  $AM_i$  yields:

$$C = \begin{bmatrix} 10 \\ 12 \\ 1 \end{bmatrix} \begin{bmatrix} 2 \\ 20 \\ 25 \end{bmatrix} \begin{bmatrix} 10 \\ 9 \\ 0 \end{bmatrix} \begin{bmatrix} 16 \\ 2 \\ 11 \end{bmatrix} \begin{bmatrix} 2 \\ 3 \\ 17 \end{bmatrix} \begin{bmatrix} 0 \\ 16 \\ 19 \end{bmatrix} .$$

Bob transmits  $C$  to Alice, and now she must recover  $M$ . Therefore, consider the inverse of  $e$ , which is

$$d = \begin{bmatrix} 21 & 10 & 12 \\ 11 & 6 & 5 \\ 17 & 22 & 21 \end{bmatrix} .$$

We multiply

$$dC = A^{-1}AM := A^{-1}AM_1 \ A^{-1}AM_2 \ \dots \ A^{-1}AM_k \ \text{mod } 27$$

which results in our original  $M$  as desired.

Ultimately, determining  $e$  from  $C$  is extremely difficult because there is no immediate intuition on how to solve for  $e$  aside from guessing, and there are approximately  $27^{n^2}$  possibilities. Thus, the security of this cryptosystem is higher than that of the Caesar Cipher since the size of the *key space* – the set of possible encryption keys – is much larger. It is still a symmetric key system, elucidating the notion that it is possible to maintain a symmetric key system and establish more security in the process.

## 2. KEY EXCHANGES

We will develop some tools that we need from Alice and Bob in order for them to agree on a secure channel. To start, we let  $G$  be a finite cyclic group of order  $n$  for the purposes of the following definitions:

**Definition 2.** The **order** of an element  $g \in G$ , denoted  $|g|$ , is the smallest positive integer  $k$  for which  $g^k = e$ , the identity element of  $G$ .

**Definition 3.** Suppose  $G = \mathbf{F}_p^\times$ , where  $p$  is a prime, and let  $b \in G$ . Then,  $b$  is a **primitive root modulo  $p$**  if  $|b| = p - 1$ .

Using these definitions, we can now develop a proposition for our group  $G$ .

**Proposition 1.** *Let  $G = \mathbf{F}_p^\times$ . If  $b$  is a primitive root modulo  $p$ , then for every  $g \in G$ , there exists some  $k \in \mathbf{N}$  such that  $g \equiv b^k \pmod{p}$ .*

*Proof.* Let  $b$  be a primitive root mod  $p$ . Then,  $|b| = p - 1$  by Definition 3. We claim that  $b^i \not\equiv b^j \pmod{p}$  for any  $1 \leq i < j \leq p - 2$ . To prove our claim, suppose that  $b^x \equiv b^y \pmod{p}$  for some  $1 \leq x \leq y \leq p - 2$ , which implies that  $b^{y-x} \equiv 1 \pmod{p}$ . We see that  $0 \leq y - x \leq p - 2$ , and since  $|b| = p - 1$  we conclude that  $y - x = 0$  which implies that  $y = x$ . Thus our claim is true, which implies that each of  $b, b^2, \dots, b^{p-2}$  is distinct. Then, as  $|b| = p - 1$ , we conclude that  $|\{b, b^2, \dots, b^{p-2}\}| = p - 1$ . Since  $G = \{1, 2, \dots, p - 2\}$  also has  $p - 1$  elements, we further conclude that  $\{b, b^2, \dots, b^{p-2}\} = \{1, 2, \dots, p - 2\}$ . Therefore for all  $g \in G$ ,  $g \equiv b^k \pmod{p}$  for some  $k \in \mathbf{N}$ , as desired.  $\square$

*Remark 1.* This proposition is equivalent to stating that a primitive root modulo  $p$  is a *generator* of its respective group.

**2.1. The Discrete Logarithm Problem.** As a motivating example, let us consider the following problem: find the  $x \in \mathbf{R}$  with  $10^x = 700$ , that is, find the logarithm  $\log_{10}(700)$ . We know that  $10^2 = 100$  and  $10^3 = 1000$ , and thus we conclude that the value of  $x$  is somewhere between 2 and 3. Further investigation by testing values such as 2.1 and 2.9 will yield more information regarding the true value of  $x$ , and ultimately we will be able to narrow down the result (which is approximately 2.845). We are able to hone in on this solution given the strictly increasing nature of exponential functions. However, this is not the case when working with our group  $G$ . We will now generalize this method to our group  $G$  and introduce the formal terminology for the Discrete Logarithm Problem (“DLP”) that arises. We define the DLP as follows:

**The Discrete Logarithm Problem:** Let  $G$  be a finite cyclic group of order  $n$ . Let  $b$  be a generator of  $G$  and  $g \in G$ . Then, calculate the unique integer  $x$ ,  $0 \leq x \leq n - 1$ , such that  $b^x = g$ ; that is, calculate the discrete logarithm  $\log_b(g)$ .

For example, suppose  $G = \mathbf{F}_p^\times$ , where for cryptographic purposes  $p$  is a large prime on the order of several hundred digits. The inherent complexity of this problem arises from the inefficiency in attempting a brute force method to find  $x$ . Due to the group’s cyclic nature, finding  $x$  does not become easier when computing consecutive powers of  $b$  in contrast to our motivating example. There are even records for developing a more efficient method to solve this problem. In fact, according to [8], the standing record for a finite field of characteristic two was set by Granger, Kleinjung, and Zumbrägel on January

31, 2014 which required 400,000 core hours of processing. Clearly, given its complexity, the DLP provides a level of security to many different cryptosystems.

We now proceed through an example of the DLP.

*Example 3.* Let  $p = 2017$ ,  $b = 423$  which is a primitive root of 2017. Find an  $x \in \mathbf{Z}$  with  $423^x \equiv 709 \pmod{2017}$ ; that is, find  $\log_{423}(709) \pmod{2017}$ , the discrete logarithm. By Proposition 1, we know that  $1 \leq x \leq 2015$ . We compute several powers of  $423 \pmod{2017}$  as follows:

$x$	$423^x$	Result mod 2017
1	$423^1$	423
2	$423^2$	1433
3	$423^3$	1059
4	$423^4$	183
5	$423^5$	763
6	$423^6$	29
$\vdots$	$\vdots$	$\vdots$
2014	$423^{2014}$	1606
2015	$423^{2015}$	1626

It is evident that in contrast to the exponential function in real numbers, the exponential function mod  $p$  is not strictly increasing or decreasing. Hence, none of these attempts give any more information about the true nature of  $x$

(which is, in fact, 1001). For much larger primes, it is infeasible to test every possible value of  $x$ , thus increasing the security of the cryptosystem.

**2.2. Diffie-Hellman Key Exchange.** It would not be useful to utilize these symmetric key cryptosystems without having a way to communicate the keys over insecure channels as well. We will now discuss methods to generate keys over those channels and focus specifically on the *Diffie-Hellman Key Exchange*. Alice and Bob agree on a prime number  $p$  (on the order of several hundred digits) and  $b$ , a primitive root mod  $p$ . Next, Alice and Bob each choose random integers  $x \bmod p$  and  $y \bmod p$ , respectively. At this point, any third party intercepting the insecure channel is aware of  $p$  and  $b$ , but only Alice knows  $x$  and only Bob knows  $y$ .

Now, Alice and Bob calculate  $A = b^x \bmod p$  and  $B = b^y \bmod p$ , respectively, and then they transmit  $A$  and  $B$  to each other. Using  $A$  and  $B$ , they will separately create a *shared symmetric key*, which will ultimately be the same for both of them. Alice calculates  $B^x \bmod p$ , and in a similar fashion, Bob calculates  $A^y \bmod p$ . The chart below details the each of these steps, demonstrating the calculations and the final result.

	Choose	Compute	Transmit	Receive	Compute
Alice	$x$	$A = b^x$	$A$	$B$	$B^x$
Bob	$y$	$B = b^y$	$B$	$A$	$A^y$

Alice has computed

$$B^x = (b^y)^x \equiv b^{xy} \pmod{p},$$



and Bob has computed

$$A^y = (b^x)^y \equiv b^{xy} \pmod{p}.$$

In summary, Alice and Bob calculated the same number  $k = b^{xy} \pmod{p}$ , which they will use as their shared symmetric key.

It is worth noting that Eve can watch these transmissions take place, learning  $p$ ,  $b$ ,  $A$ , and  $B$ , but in order to use those to determine  $k$ , she also needs  $x$  or  $y$ . To solve for  $x$  or  $y$ , she would need to calculate either  $\log_b(A) \pmod{p}$  or  $\log_b(B) \pmod{p}$  which would in turn solve a new, more general version of the DLP defined as follows:

**The Diffie-Hellman Problem (“DHP”):** Let  $G$  be a finite cyclic group of order  $n$ . Let  $b$  be a generator of  $G$ , and let  $b^x, b^y$ , and  $g \in G$  be given to the attacker. Then, the attacker must calculate the unique integer  $xy$ ,  $0 \leq xy \leq n - 1$ , such that  $b^{xy} = g$ ; that is, calculate the discrete logarithm  $\log_b(g)$ .

As we discussed in Section 2.1, calculating these logarithms mod  $p$  is generally not thought to be feasible within a reasonable amount of time. Therefore, as long as  $p$  was chosen well, Alice and Bob have securely agreed on a key over an insecure channel. However, as we pointed out there are some attacks such as the Index Calculus Algorithm which sometimes make it possible to solve the DLP in an efficient amount of time.

The professional consensus regarding the difficulty of the DHP is that it is about as difficult as the DLP given their similar setup; however, one more

piece of information is present. Though that may seem to imply that the DHP is easier, it is still expected to be just as hard.

**2.3. The Index Calculus Algorithm.** As example 3 demonstrates, brute force is not very efficient against the DLP, but there are slightly better ways to attack this problem. There exist several algorithms that can be used to attempt to compute the discrete logarithms. Some of the notable ones are the Baby-step, Giant-step algorithm, Pollard's  $\rho$  algorithm, the Pohlig-Hellman algorithm, and the Index Calculus algorithm. We focus on some of the details of the last one, the Index Calculus algorithm, to motivate our further discussion. In order to execute the algorithm, we require two definitions and a corollary.

**Definition 4.** Let  $B \in \mathbf{Z}_+$ . We call  $m \in \mathbf{Z}_+$   **$B$ -smooth** if no prime factor of  $m$  exceeds  $B$ .

*Example 4.* To demonstrate definition 4, consider  $m = 130$ . The prime factors of 130 are 2, 5, and 13. 130 is 13-smooth since no prime factor of 130 exceeds 13, but it is not 7-smooth since 13 is greater than 7.

**Definition 5.** Let  $p_1, p_2, \dots, p_n$  be small prime numbers. Then, a **factor base** is a set  $F = \{p_1, p_2, \dots, p_n\}$ .

*Example 5.* Let  $n = 5$ . Then, a factor base  $F = \{2, 5, 29, 11, 997\}$ .

Next, recall that for  $p$ , a prime and  $b$ , a primitive root mod  $p$ ,

- $\log_b(y) \equiv x \pmod{p-1}$  if and only if  $b^x \equiv y \pmod{p}$ ,
- $\log_b(a_1 a_2) \equiv \log_b(a_1) + \log_b(a_2) \pmod{p-1}$ , and

- $\log_b(a^k) \equiv k \log_b(a) \pmod{p-1}$ .

**Corollary 1.** *If  $y$  has the prime factorization  $y = p_1^{e_1} p_2^{e_2} \dots p_t^{e_t}$ , then*

$$\log_b(y) \equiv e_1 \log_b(p_1) + e_2 \log_b(p_2) + \dots + e_t \log_b(p_t) \pmod{p-1}.$$

The proof of Corollary 1 is direct from what we recalled prior to stating it. The general process of Index Calculus algorithm is that if we can compute  $\log_b(p_i)$  for  $i = 1, \dots, t$  in a factor base, then we can assemble each individual logarithm to calculate the discrete logarithm of  $y$ . There are two phases to the algorithm: in Phase 1 we find the discrete logarithms for a list of small primes, and then in Phase 2 we assemble the discrete logarithms to obtain  $\log_b(y)$ .

**Phase 1 - Step 1:** For random  $e \in \mathbf{Z}$  with  $1 \leq e < p-1$ , compute  $y = b^e \pmod{p}$  and then factor  $y$  completely. Check if  $y$  has any prime factors other than those in  $F = \{p_1, \dots, p_t\}$ ; if so, they might not be  $y$ -smooth, so discard them. Then, find  $e_1, \dots, e_t$  such that  $y = p_1^{e_1} p_2^{e_2} \dots p_t^{e_t}$ . Note that  $\log_b(b^e) \equiv e \pmod{p-1}$ , and then calculate the base  $b$  logarithm of both sides. Then, since  $y = b^e$ , we obtain  $e \equiv e_1 \log_b(p_1) + e_2 \log_b(p_2) \dots + e_t \log_b(p_t) \pmod{p-1}$ . Repeat this process until at least  $t$  congruences involving  $\log_b(p_1), \dots, \log_b(p_t)$  are obtained.

**Phase 1 - Step 2:** Simultaneously solve all of the congruences. Then, we know the values of  $\log_b(p_1), \dots, \log_b(p_t)$ .

**Phase 2:** Recall our friends Alice, Bob, and Eve from earlier. Eve intercepts  $X$ , a transmission from Alice to Bob during the Diffie-Hellman Key Exchange.

Eve also possesses their values of  $p$  and  $b$ . Eve will now compute  $x = \log_b(X) \pmod{p-1}$ . If  $X$  factors over  $p_1, \dots, p_t$ , then Eve can decipher the message. If not, she can attempt multiplying  $X$  by various  $b^e \pmod{p}$  until  $Xb^e \pmod{p}$  factors over  $p_1, \dots, p_t$  as follows:

**Step 1:** For random  $f \in \mathbf{Z}$  with  $1 \leq f \leq p-2$ , compute  $z = b^f X \pmod{p}$  until an  $f$  satisfies  $z = p_1^{f_1} p_2^{f_2} \dots p_t^{f_t}$ .

**Step 2:** Calculate the base  $b$  logarithm of both sides, i.e.,

$$\log_b(b^f X) = \log_b(b^f) + \log_b(X) \equiv f + \log_b(X) \pmod{p-1}.$$

Also,

$$\log_b(b^f X) = f_1 \log_b(p_1) + \dots + f_t \log_b(p_t),$$

and thus,

$$\log_b(X) \equiv f_1 \log_b(p_1) + \dots + f_t \log_b(p_t) - f \pmod{p-1}.$$

We proceed through an example that demonstrates the use of the Index Calculus Algorithm:

*Example 6.* Let  $p = 131$ ,  $b = 2$ , and let the factor base be  $\{2, 3, 5, 7\}$ . Eve would like to compute  $\log_2(37)$ . We begin with Phase 1 in which we must calculate  $\log_2(2)$ ,  $\log_2(3)$ ,  $\log_2(5)$ , and  $\log_2(7)$ .

**Step 1:** We want to find values of  $e \in \mathbf{Z}$  such that  $2^e \pmod{131}$  factors over  $\{2, 3, 5, 7\}$ . We choose values of  $e$  randomly and find that  $2^1 \equiv 2$ ,  $2^8 \equiv 5^3$ ,  $2^{12} \equiv (5)(7)$ ,  $2^{14} \equiv 3^2$ , and  $2^{34} \equiv (3)(5^2)$ . We take the logarithms of these

values mod 130 and find five initial congruences:  $1 \equiv \log_2(2)$ ,  $8 \equiv 3 \log_2(5)$ ,  $12 \equiv \log_2(5) + \log_2(7)$ ,  $14 \equiv 2 \log_2(3)$ , and  $34 \equiv \log_2(3) + \log_2(5)$ .

We note that  $\log_2(2) \equiv 1$  is obvious. Then, for  $8 \equiv 3 \log_2(5) \pmod{130}$ , we compute  $3^{-1} \pmod{130} = -43$ . So  $(-43)(8) \equiv (-43)(3) \log_2(5) \pmod{130}$ ,  $(-43)(8) \equiv \log_2(5) \pmod{130}$ ,  $-344 \equiv \log_2(5) \pmod{130}$ , and therefore  $46 \equiv \log_2(5) \pmod{130}$ .

Next,  $12 \equiv \log_2(5) + \log_2(7) \pmod{130}$ . By the second initial congruence,  $12 \equiv 46 + \log_2(7) \pmod{130}$ . So  $-34 \equiv \log_2(7) \pmod{130}$  and thus  $96 \equiv \log_2(7) \pmod{130}$ .

Next,  $14 \equiv 2 \log_2(3) \pmod{130}$ . **Note:** at an initial glance, it may appear that  $\log_2(3) \equiv 7 \pmod{130}$ , but this is not true. The reason is due to the fact that  $b \equiv ax \pmod{n}$  has  $\gcd(a, n)$  solutions, and here,  $\gcd(2, 130) = 2$ .

Lastly, we have  $34 \equiv \log_2(3) + \log_2(5) \pmod{130}$ , which is congruent to  $\log_2(3) + (2)(46) \pmod{130} \equiv \log_2(3) + 92 \pmod{130}$ . Thus,  $\log_2(3) \equiv 72 \pmod{130}$ . We verify that  $(2)(72) \equiv 144 \equiv 14 \pmod{130}$ , which is the same as our result from the fourth initial congruence. Therefore,  $\log_2(2) \equiv 1$ ,  $\log_2(3) \equiv 72$ ,  $\log_2(5) \equiv 46$ , and  $\log_2(7) \equiv 96$ .

We proceed to **Phase 2:** Eve wishes to calculate  $\log_2(37) \pmod{130}$ . She attempts to calculate  $b^f(37) \pmod{131}$  until the resulting number factors are  $\{2, 3, 5, 7\}$ . She ultimately finds that  $2^{43}(37) \equiv 105 \equiv (3)(5)(7) \pmod{131}$ .

Then,  $43 + \log_2(37) \equiv \log_2(3) + \log_2(5) + \log_2(7) \pmod{130}$ , which implies that  $\log_2(37) \equiv 72 + 46 + 96 - 43 \pmod{130}$ . Therefore, we conclude that  $\log_2(37) \equiv 41 \pmod{130}$ . We verify this by computing  $2^{41} \pmod{131}$ , which is equal to 37, as desired.

Though this method works in this setting, there are limitations in that the factor base needs to have enough primes so that many numbers factor over it. However, that causes a direct increase in the required number of congruences. Additionally, solving congruences simultaneously is difficult, which can cause the algorithm to take longer. Most importantly, difficulties arise with the Index Calculus algorithm when working with certain objects called elliptic curves over finite fields. The concept of smoothness fails to apply because their structure does not allow for decomposition into prime divisors, and thus the algorithm fails as well. Therefore, we wish to use elliptic curves for more secure cryptographic purposes, but we require a generalization of the DLP in order to do so.

### 3. ELLIPTIC CURVES

**3.1. Introduction.** As a brief overview, there are other environments in which we can simultaneously increase the security of our cryptosystem and the efficiency of the encryption and decryption processes [5]. Specifically, we will apply the concept of the Diffie-Hellman Key Exchange to a setting that employs elliptic curves. Formally, an elliptic curve over a field  $K$  is defined as a smooth, genus one projective curve of the form

$$E : y^2 = x^3 + ax + b, \quad a, b \in K.$$

We will delve further into the details of this definition and establish a foundation upon which we can apply the Diffie-Hellman Key Exchange. As a precursor, we note that Silverman and Tate's text [4] is the primary source referenced for this material.

**3.2. Elliptic Curves and a Group Law.** We begin by defining the **Euclidean (or affine) plane** for a field  $K$  as

$$\mathbf{A}^n(K) = \{(x_1, \dots, x_n) : x_i \in K\}.$$

Then, we define an equivalence relation on  $\mathbf{A}^n(K)$  as follows:

**Definition 6.** Two points of  $\mathbf{A}^n(K)$  are **projectively equivalent**, denoted  $[x_1, \dots, x_n] \sim [y_1, \dots, y_n]$ , if there exists a  $\lambda \neq 0$ ,  $\lambda \in K$ , with  $x_i = \lambda y_i$  for  $i = 1, \dots, n$ .

We now use this concept to define the space in which elliptic curves exist.

**Definition 7.** We define **projective space** as

$$\mathbf{P}^n(K) = \frac{\mathbf{A}^{n+1}(K) \setminus \{0\}}{\sim}.$$

Specifically, we work within  $\mathbf{P}^2(K)$  for the remainder of this paper. Consider the points in  $\mathbf{P}^2(K)$  which satisfy the following equation:

$$Y^2Z = X^3 + aXZ^2 + bZ^3, \quad \text{where } a, b \in \mathbf{Q} \text{ and } 4a^3 + 27b^2 \neq 0.$$

(We note here that since we have placed this constraint on the coefficients, we have guaranteed that the curve is nonsingular.) Since there is always a nonzero coordinate by definition, it is always possible to divide each of the

coordinates by that nonzero coordinate. More precisely, we have a projective equivalence between the points  $[x, y, z] \in \mathbf{P}^2(K)$  and  $[\frac{x}{z}, \frac{y}{z}, 1]$ . It is then simple to show that we can identify the set of projective points with a one in a single coordinate to the affine n-space.

Given these facts, observe that if we substitute  $Z = 0$  into this equation, then the equation becomes  $0 = X^3$  and we obtain exactly one projective point  $[0, Y, 0] \sim [0, 1, 0]$ . This point is called the *point at infinity*, denoted  $\mathcal{O}$ , and we will soon discover its significance for our discussion. This conclusion exhausts the case where  $Z = 0$ .

Next, consider the case where  $Z \neq 0$ , or equivalently,  $Z = 1$ . We also set  $\frac{X}{Z} = x$  and  $\frac{Y}{Z} = y$  in our equation, yielding

$$y^2 = x^3 + ax + b.$$

Cubics of this form are identified as being in *Weierstrass normal form*, and we work exclusively with these kinds of cubics for the remainder of this paper. We have exhausted the possible values of  $Z$ , and finally, we can define a general elliptic curve  $E$  as follows:

**Definition 8.** An **elliptic curve**  $E$  over  $\mathbf{Q}$  is the curve defined by an equation of the form

$$E : Y^2Z = X^3 + aXZ^2 + bZ^3, \quad \text{where } a, b \in \mathbf{Q} \text{ and } 4a^3 + 27b^2 \neq 0.$$

The **K-rational points on**  $E$ , denoted  $E(K)$ , are therefore  $\mathcal{O}$  and the points in  $\mathbf{A}^2(K)$  satisfying  $y^2 = x^3 + ax + b$ .



Now we are ready to visualize an elliptic curve (all of the following figures were produced manually using [1]). Figure 3.1 is an example of the real points on an elliptic curve, denoted  $E(\mathbf{R})$ , and we think of  $\mathcal{O}$  as living infinitely far up the  $y$ -axis (depicted by the arrow).

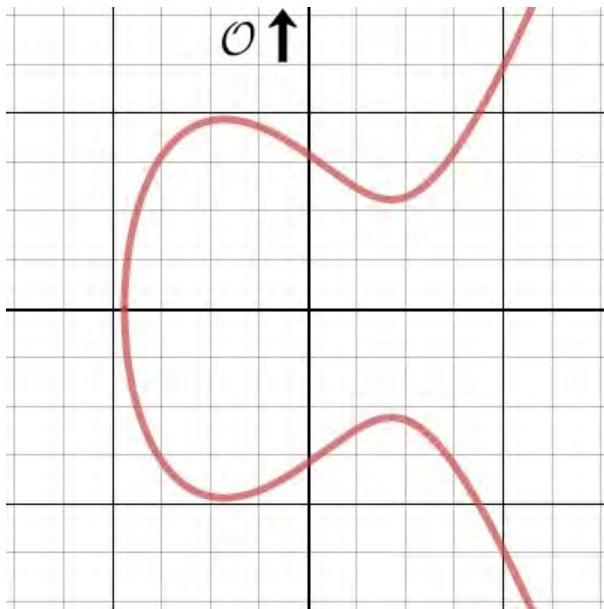


FIGURE 3.1.  $E(\mathbf{R})$

Next, assume that we have an elliptic curve  $E$  and two rational points  $P$  and  $Q$ . How can we generate more rational points other than the two that we already know? The answer is surprisingly geometric in nature. Consider Figure 3.2 that depicts the location of these two points.

First, we construct a line through  $P$  and  $Q$ , extending through the graph at a third point of intersection.<sup>1</sup> We name this intersection  $P*Q$ , which is shown in Figure 3.3. Next, we construct a line through  $P*Q$  and  $\mathcal{O}$ , which is actually

<sup>1</sup>It is simple to show that we can always locate a third point of intersection, but we will discuss more details in the next proof.

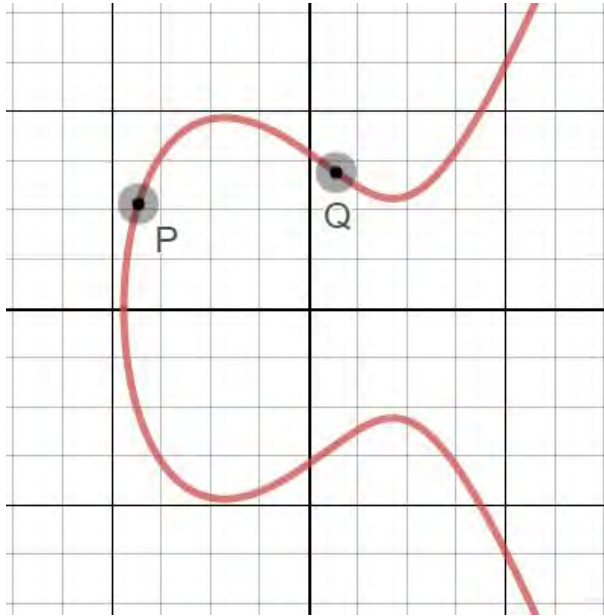


FIGURE 3.2.  $E(\mathbf{R})$  with Points  $P$  and  $Q$

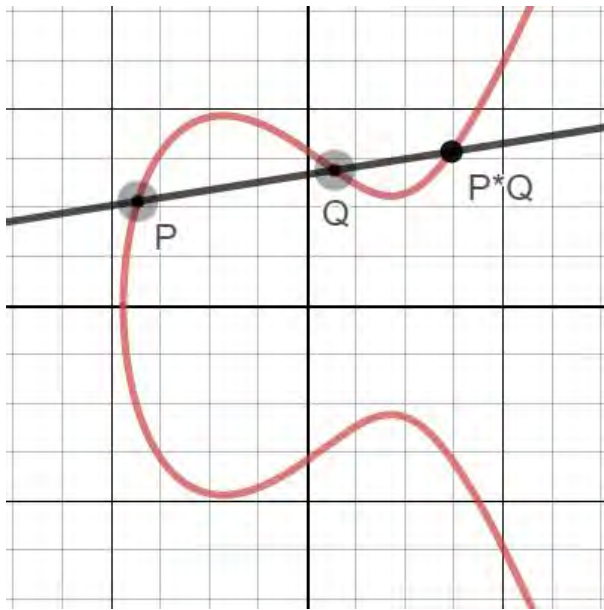


FIGURE 3.3. The Line Joining  $P$  and  $Q$

the vertical line through  $P * Q$  since we view  $\mathcal{O}$  as being infinitely far up the  $y$ -axis. In fact, this vertical line passes through  $P * Q$ ,  $\mathcal{O}$ , and the elliptic curve

in the third point. Since cubics in Weierstrass form are symmetric about the  $x$ -axis, we can reflect  $P*Q$  over the  $x$ -axis, resulting in a fourth point that we name  $P + Q$ . Lastly, we define the negative of an arbitrary point  $P$  to be the

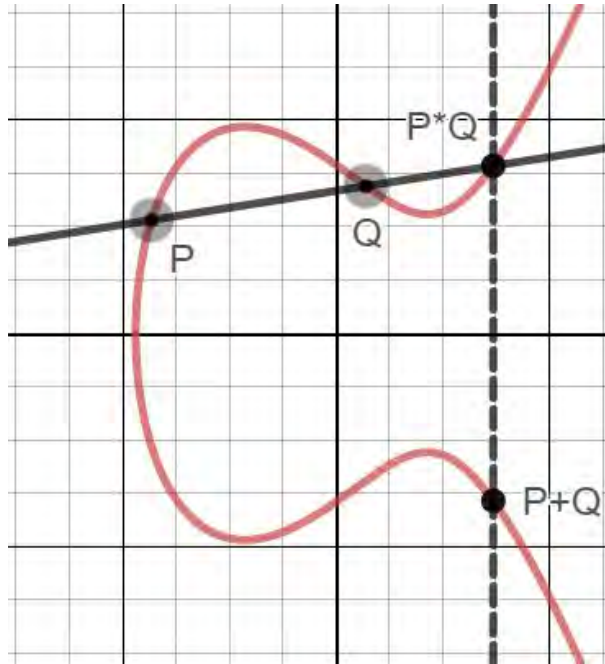


FIGURE 3.4. The Reflection for  $P + Q$

point obtained  $P$  is reflected over the  $x$ -axis. Observe that if we add  $P$  and  $-P$ , we obtain  $\mathcal{O}$  as depicted in Figure 3.5.

The third point of intersection between  $P$  and  $\mathcal{O}$  is actually  $-P$ , and in fact, it is in Figure 3.5. To add  $P$  and  $-P$ , we construct the line between them, which is vertical. The line's third point of intersection is  $\mathcal{O}$ , and connecting  $\mathcal{O}$  with itself yields  $\mathcal{O}$  again. Thus, we have that  $P + -P = \mathcal{O}$ , and therefore,  $P$ 's negation is  $-P$ . This definition may cause suspicion of a group structure, so we continue with the following theorem that relates  $E$  and  $+$ .

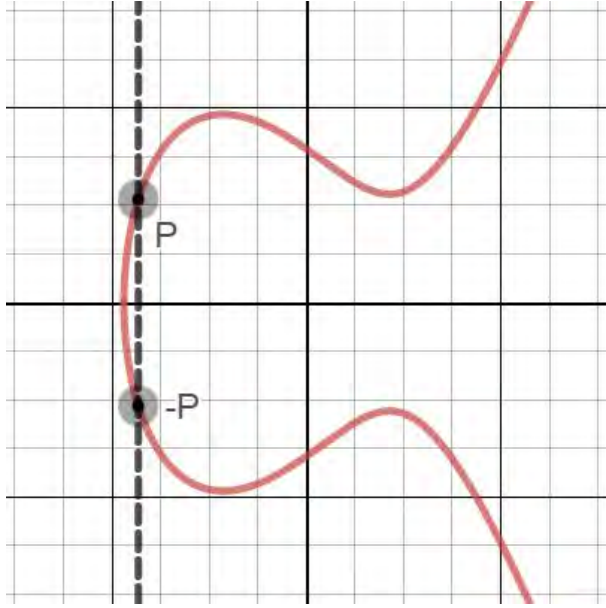


FIGURE 3.5.  $P$  and  $-P$

**Theorem 1.** Let  $E$  be an elliptic curve and  $+$  be the operation defined above.

Then, for all  $P, Q, R$  in  $E(K)$ ,

- i.  $P + Q = Q + P$
- ii.  $P + \mathcal{O} = \mathcal{O} + P = P$
- iii.  $P + -P = \mathcal{O}$
- iv.  $(P + Q) + R = P + (Q + R)$

Thus,  $(E, +)$  forms an abelian group.

*Proof.* We proceed through the proof geometrically.

i. We determined our points of intersection by constructing a line through  $P$  and  $Q$  and there is only one such line; therefore it is immediate that  $P + Q = Q + P$ .

ii. To add  $P$  and  $\mathcal{O}$ , we construct a line from  $P$  to  $\mathcal{O}$  and obtain  $P * \mathcal{O}$ . However, the line connecting  $P$  and  $\mathcal{O}$  is simply a vertical line through  $P$ , and

thus we obtain  $P$  again, so  $P * \mathcal{O} = P$ . Connecting it again yields the same result, so it follows that  $P + \mathcal{O} = \mathcal{O} + P = P$ .

*iii.* This is immediate when considering the definition of  $P$  and  $-P$ .

*iv.* In order to prove associativity, we require a definition, a significant theorem, and a corresponding lemma. We develop these separately from this proof since they will be lengthy.

□

**Definition 9.** A **projective curve**  $C$  is the set of solutions to a projective equation

$$C : F(X, Y, Z) = 0$$

where  $F$  is a non-constant polynomial, each of whose monomials is of the same degree,  $d$ . We call  $d$  the **degree** of  $C$ . For example, if  $C : Y^2 + X^2 + XZ^3$ , then the degree of  $C$  is 3.

**Theorem 2. Bézout's Theorem:** *Let  $C$  and  $D$  be two projective curves of degrees  $m$  and  $n$ , respectively. Assume that  $C$  and  $D$  do not intersect at infinitely many points. Then,  $C$  and  $D$  intersect at  $mn$  points.*

We accept this theorem without proof. Since the elliptic curves that we are working with have degree three, Bézout's Theorem implies that any two of them will intersect at nine distinct points. We also highlight that some of the points can have a multiplicity greater than one, so some may be counted multiple times, and  $\mathcal{O}$  is one of these points since all of the curves include it. Now, we can apply Bézout's Theorem to develop a lemma (which is actually the Cayley-Bacharach Theorem<sup>2</sup>) that will aid us in proving associativity.

<sup>2</sup> For more information, refer to [4, pg 240].

**Lemma 1 (The Cayley-Bacharach Theorem).** *Let  $C$ ,  $D$ , and  $E$  be three elliptic curves in projective space that do not intersect at infinitely many points. Then, if  $E$  shares eight of the nine points of intersection between  $C$  and  $D$ , then  $E$  also shares the ninth point of intersection.*

Now, we are ready to return to the last part of Theorem 1 and prove associativity of points on elliptic curves.

*Proof of iv.* First, let  $P$ ,  $Q$ , and  $R$  be three arbitrary points on an elliptic curve  $E$ . We want to show  $P + (Q + R) = (P + Q) + R$ , but that can also be achieved by showing  $P * (Q + R) = (P + Q) * R$  because the final step is simply reflecting both points over the  $x$ -axis. Now, let  $L_1$  be the line that connects  $P$ ,  $Q$ , and  $P * Q$ ; let  $L'_1$  be the line that connects  $Q$ ,  $R$ , and  $Q * R$ ; let  $L_2$  be the vertical line that connects  $\mathcal{O}$ ,  $Q * R$  and  $Q + R$ ; let  $L'_2$  be the vertical line that connects  $\mathcal{O}$ ,  $P * Q$ , and  $P + Q$ ; let  $L_3$  be the line that connects  $P + Q$  and  $R$ ; lastly, let  $L'_3$  be the line that connects  $P$  and  $Q + R$ . Since  $E$  is in projective space,  $L_3$  and  $L'_3$  intersect at exactly one point, which we name  $X$ . Then, since  $L_3$  and  $L'_3$  are lines that connect two points on  $E$ , by Bézout's Theorem, they intersect  $E$  at a third point as well. Therefore, if we can show  $X$  is on  $E$ , then we have  $X = P * (Q + R) = (P + Q) * R$  as desired.

Let  $S = \{\mathcal{O}, P, Q, R, P * Q, Q * R, P + Q, Q + R, X\}$ . From our definitions of each line, every point in  $S$  has two lines  $L_i$  and  $L'_i$  that intersect it. Let  $C = L_1 \cup L_2 \cup L_3$  and  $D = L'_1 \cup L'_2 \cup L'_3$ . Since each  $L_i$  and  $L'_i$  is a degree one projective curve,  $C$  and  $D$  are projective curves of degree 3. By their construction,  $C$  and  $D$  intersect every point in  $S$ . Since every point in  $S$

aside from  $X$  was defined to be on  $E$ , we now have that  $E$  shares eight of the nine points of intersection between  $C$  and  $D$ . Therefore, by Lemma 1,  $E$  also shares the ninth point of intersection,  $X$ . Therefore,  $P*(Q+R) = (P+Q)*R$ , and we have proven associativity, as desired. Figure 3.6 depicts the entirety of this proof.

□

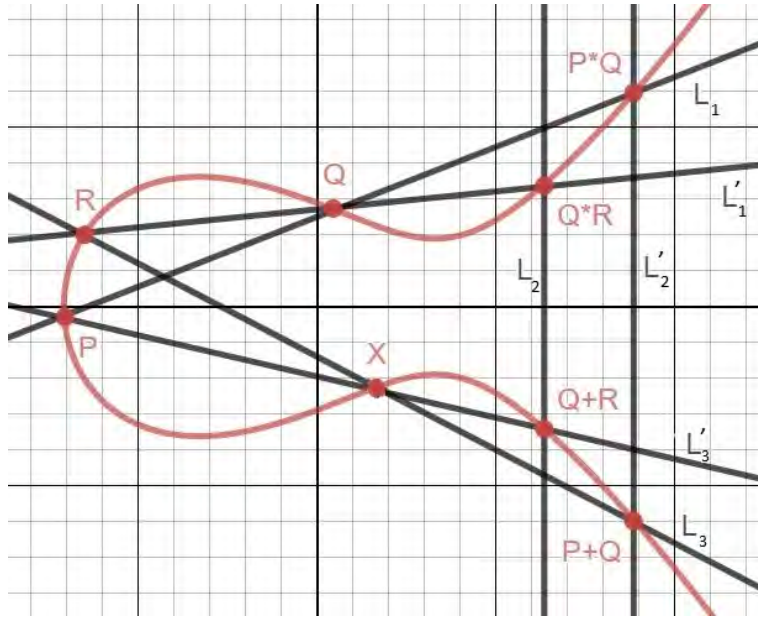


FIGURE 3.6.  $E(\mathbf{R})$ ,  $S$ , the  $L_i$ , and the  $L'_i$

Since we cannot produce exact values using only geometry, we will develop formulas for computing the addition of these points. As Silverman and Tate [4] explain, we let  $P_1 = (x_1, y_1)$ ,  $P_2 = (x_2, y_2)$ ,  $P_1 * P_2 = (x_3, y_3)$ , and  $P_1 + P_2 = (x_3, -y_3)$ . Given  $P_1$  and  $P_2$ , we would like to compute  $P_1 + P_2$ , so consider the equation of the line connecting them. This line can be defined as having an equation

$$y = \lambda x + \nu, \text{ with } \lambda = \frac{y_2 - y_1}{x_2 - x_1} \text{ and } \nu = y_1 - \lambda x_1 = y_2 - \lambda x_2.$$

We now substitute

$$y^2 = (\lambda x + \nu)^2 = x^3 + ax + b,$$

and move all of the terms to the right side of the equation, yielding

$$0 = x^3 - \lambda^2 x^2 + (a - 2\lambda\nu)x + (b - \nu^2).$$

This equation is that of a cubic in  $x$ , and we can compute the coordinates of the three intersections by computing the roots  $x_1$ ,  $x_2$ , and  $x_3$ . In doing so, we obtain

$$x^3 - \lambda^2 x^2 + (a - 2\lambda\nu)x + (b - \nu^2) = (x - x_1)(x - x_2)(x - x_3),$$

and by expanding the right side, we have

$$x^3 - \lambda^2 x^2 + (a - 2\lambda\nu)x + (b - \nu^2) = x^3 - (x_1 + x_2 + x_3)x^2 + \dots = 0.$$

We equate the coefficients of the  $x^2$  term on both sides, resulting in the equation  $\lambda^2 = x_1 + x_2 + x_3$ , and obtain

$$x_3 = \lambda^2 - x_1 - x_2. \tag{1}$$

Then, we use Equation 1 in the point-slope form of the line to obtain

$$y_3 - y_1 = \lambda(x_3 - x_1)$$



and therefore, by subtraction,

$$y_3 = \lambda(x_3 - x_1) + y_1. \quad (2)$$

Thus, we have Equation 1 and 2 for computing the sum of two arbitrary points  $P_1$  and  $P_2$  on an elliptic curve. We proceed through an example to use these formulae.

*Example 7.* Let  $E$  be given by  $y^2 = x^3 + 17$ . Let  $P = (-2, 3)$  and  $Q = (-1, 4)$ , which are both on  $E$ . Find  $P + Q$ .

**Solution:** First, it is clear that  $E$  satisfies our general elliptic curve form. Next, we calculate

$$\lambda = \frac{y_2 - y_1}{x_2 - x_1} = \frac{4 - 3}{-1 - (-2)} = \frac{1}{1} = 1.$$

Using Equation 1 and 2, respectively, we calculate

$$x_3 = \lambda^2 - x_1 - x_2 = 1^2 - (-2) - (-1) = 4;$$

$$y_3 = \lambda(x_3 - x_1) + y_1 = (1)(4 - (-2)) + (3) = 9.$$

Thus, we have that  $P + Q = (-2, 3) + (-1, 4) = (4, 9)$ , which satisfies  $y^2 = x^3 + 17$ .

We should also consider the case of adding a point to itself; however, since it involves the slope of a tangent line, we need to modify our equations. First, let  $P = (x, y)$ . We want to calculate  $P + P = 2P$ , so we construct the line that connects  $P$  to itself which is simply the tangent line to the curve at  $P$ . We express the curve in the relation  $y^2 = f(x)$ , and through implicit

differentiation, we obtain:

$$\lambda = \frac{dy}{dx} = \frac{f'(x)}{2y}.$$

We can now substitute  $\lambda$  into Equation 1 and 2 to obtain the result of  $P + P = 2P$ . It is common to call the following formula for this process the *duplication formula*:

$$\text{The } x \text{ value of } 2P = \frac{x^4 - 2ax^2 - 8bx + a^2}{4x^3 + 4ax + 4b}. \quad (3)$$

Now that we can add and double these points, we have enough tools to generate the relevant groups of rational points.

*Remark 2.* It is essential to note that all of the computations we just completed still hold when working algebraically mod an odd prime  $p$ . The differences, however, are that we can no longer view the geometry of our curves and the new constraint on the coefficients is

$$4a^3 + 27b^2 \not\equiv 0 \pmod{p}. \quad (4)$$

We shall proceed through an example to demonstrate the use of these equations.

*Example 8.* Consider  $E : y^2 = x^3 + 2x + 2 \pmod{17}$  and  $P = (5, 1)$ . Compute  $2P$ .

Here, we see that  $p = 17$ ,  $a = b = 2$ ,  $x = 5$ , and  $y = 1$ . First, we verify that the curve is smooth using Equation (4). We evaluate  $4a^3 + 27b^2 = 4(2)^3 + 27(2)^2 = 32 + 108 = 140$ , and  $140 \pmod{17} \equiv 4 \not\equiv 0$  as desired, so  $E$  is smooth. Now we want to compute  $2P = P + P = (5, 1) + (5, 1)$ , so we apply the duplication formula (Equation (3)) as follows:

$$\begin{aligned}
x \text{ value of } 2P &= \frac{5^4 - (2)(2)(5)^2 - (8)(2)(5) + (2)^2}{(4)(5)^3 + (4)(2)(5) + (4)(2)} \pmod{17} \\
&= \frac{625 - 100 - 80 + 4}{500 + 40 + 8} \pmod{17} \\
&= \frac{7}{4} \pmod{17} \\
&= 7(4^{-1}) \pmod{17} \\
&= 7(-4) \pmod{17} \\
&= -28 \pmod{17} \\
&= 6.
\end{aligned}$$

Now that we have the  $x$ -value of  $2P$ , we can simply use  $E$  to determine the corresponding  $y$ -value. Using the slope of the tangent line, we evaluate  $E$  at  $x = 6$  and solve for  $y$  to obtain  $y = 3$ . Thus, we have that  $2P = (5, 1) + (5, 1) = (6, 3)$ .

Now that we know how to add both distinct and non-distinct points, it is worth mentioning that there is a process that can help us calculate large multiples of  $P$  even more efficiently. The **Square-and-Multiply Algorithm** essentially allows us to use the duplication formula repeatedly until we obtain the multiple that we want.

For example, if we wish to calculate  $9P$ , then we can also view this as a sum of several iterations of the duplication formula and one addition, i.e.,  $9P = P + 2(2(2P))$ . We already know how to duplicate a point, so we duplicate  $P$  to get  $2P$ ,  $2P$  to get  $2(2P) = 4P$ , and finally  $2(2(2P))$  to get

$2(4P) = 8P$ . Then, simply add  $P$  and  $8P$  and we obtain  $9P$  as desired. The advantage to using such an algorithm arises in that each multiple of any point can be calculated by repeated duplications and one addition.

**3.3. DHP and Elliptic Curves.** We can now combine the DHP discussed earlier with the elliptic curve setting. Consider the following definition:

**The Elliptic Curve Discrete Logarithm Problem (“ECDLP”):** Let

$$E : y^2 = x^3 + ax + b, \quad a, b \in \mathbf{Z}$$

be an elliptic curve, and as before write  $E(\mathbf{F}_p) = \mathbf{A}^2(\mathbf{F}_p) \cup \{\mathcal{O}\}$  for the  $\mathbf{F}_p$ -rational points on  $E$ . Let  $P, Q \in E(\mathbf{F}_p)$  where  $Q$  is some multiple of  $P$ . Then, calculate the value of  $n \in \mathbf{Z}$  such that  $nP = Q$ , i.e., calculate the  $n$  number of times that  $P + P + \dots + P = Q$ , which is a discrete logarithm over  $E$ , denoted  $k = \log_P Q$ .

We call on our old friends Alice and Bob to demonstrate the design of a cryptosystem using the ECDLP. Alice and Bob mutually choose a prime  $p$ , an elliptic curve  $E$ , and a point  $P \in E(\mathbf{F}_p)$ . Alice chooses a random integer  $a$ , calculates  $P_a = aP$ , and sends  $P_a$  to Bob. Simultaneously, Bob chooses a random integer  $b$ , calculates  $P_b = bP$ , and sends  $P_b$  to Alice. Alice receives  $P_b$  and then multiplies it by  $a$  to obtain  $abP$ , and Bob receives  $P_a$  and multiplies it by  $b$  to obtain  $abP$  as well since the group law on  $E$  is abelian. Thus, the two have constructed a shared secret key. This scenario is depicted in the following table:

	Choose	Compute	Transmit	Receive	Compute	Result
Alice	$a$	$aP$	$P_a$	$P_b$	$aP_b$	$abP$
Bob	$b$	$bP$	$P_b$	$P_a$	$bP_a$	$abP$

Now, our third party Eve cannot compute  $abP$  without being able to solve the ECDHP. Accomplishing such a task is thought to be quite difficult and is far more computationally demanding than the previous problems that we have discussed.

An important strength to using elliptic curves for cryptographic purposes is that the Index Calculus algorithm (previously described in section 2.3) is not applicable for most elliptic curves. In addition, according to [5], it is commonly thought that solving the discrete logarithm problem in  $E(\mathbf{F}_p)$  is significantly more difficult than in  $(\mathbf{Z}/p\mathbf{Z})^\times$ . This fact promotes cryptosystems that employ elliptic curves because the values that the two parties must use can be a lot smaller. Rather than working with primes on the order of 300 digits and then performing calculations with numbers of comparable size, it is feasible to use numbers that are far smaller.

#### 4. CRYPTANALYSIS: ATTACKS AGAINST THE ECDLP

Using elliptic curves for cryptographic purposes offers significant advantages, and as we mentioned in Section 2.3, the Index Calculus algorithm fails to work for them. We cannot develop an equivalent concept to smoothness for all elliptic curves as we cannot satisfy the assumption of Corollary 1 (that is, the capability of having a prime factorization). Still, though there are benefits to using elliptic curves for cryptography, there are other cryptanalytic attacks that can be executed to attempt to discern the value of  $n$  in the ECDLP. We now focus on a few choices of these attacks to demonstrate their capabilities

and strengths in intercepting messages used by the cryptosystems that we have discussed.

4.1. **The Baby-Step, Giant-Step Algorithm.** Before we begin, we require one more definition regarding elliptic curves.

**Definition 10.** The **number** of  $\mathbf{F}_p$ -rational points on an elliptic curve  $E$  is denoted  $\#E(\mathbf{F}_p)$ .<sup>3</sup>

The Baby-Step, Giant-Step algorithm can be applied to any arbitrary group, and we apply it to the elliptic curve environment.

**Problem:** Let  $kP = Q$  in  $E(\mathbf{F}_p)$  and  $\#E(\mathbf{F}_p) = N$ . Calculate  $k$ .

The Baby-Step, Giant-Step algorithm proceeds as follows:

- (1) Choose  $m \in \mathbf{Z}$  such that  $m > \sqrt{N}$ .
- (2) Calculate  $mP$ .
- (3) For all  $0 \leq a \leq m - 1$ , calculate and record  $aP$  into a list. Similarly, for all  $0 \leq b \leq m - 1$ , calculate and record  $Q - bmP$  into a second list.
- (4) Compare both lists for the values of  $aP$  and  $Q - bmp$  until a match is found.
- (5) Once a match is found, calculate  $k \equiv a + bm \pmod{N}$ .

Though this algorithm is generally regarded as one of the fastest, it is not considered feasible to use for all elliptic curves (which we will discuss in Section

---

<sup>3</sup> There is not an explicit formula to calculate this number, but it can be estimated — that is outside the scope of this paper.

4.3). We proceed through an example from Washington [6] to demonstrate its use.

*Example 9.* Consider  $E : y^2 = x^3 + 2x + 1 \pmod{41}$  and  $\#E(\mathbf{F}_{41}) \leq 54$ . We have that  $P = (0, 1)$ ,  $Q = (30, 40)$ , and  $kP = Q$ . We will now calculate  $k$ .

- (1) We choose  $m = 8$  because  $8 > \sqrt{54}$ .
- (2) We calculate  $8P = 8(0, 1) = (10, 18)$ .
- (3) For all  $0 \leq a \leq 7$ , we calculate and record  $aP$  into a list as follows:

$a$	$aP$
1	(0, 1)
2	(1, 39)
3	(8, 23)
4	(38, 38)
5	(23, 23)
6	(20, 28)
7	(26, 9)

Similarly, for  $0 \leq b \leq 7$ , we calculate and record  $Q - bmP$  into a second list as follows:

$b$	$Q - bmP$
0	(30, 40)
1	(9, 25)
2	(26, 9)

We stop at  $b = 2$  because we see that the third entry in the second list matches the seventh entry in first list. Therefore,  $a = 7$  and  $b = 2$ .

(4) We calculate

$$\begin{aligned} Q &= (a + bm)P \\ &= (7 + 2(8))P \\ &= 23P. \end{aligned}$$

Thus,  $k = 23$ .

(5) Lastly, we verify that  $23P = 23(0, 1) = (30, 40)$ .

A great advantage to the Baby-Step, Giant-Step algorithm is its *deterministic* nature, that is, it is guaranteed to find a solution.

**4.2. Pollard's  $\rho$  Algorithm.** We can apply Pollard's  $\rho$  algorithm to any cyclic group  $G$ , but we will focus specifically on the  $\mathbf{F}_p$ -rational points on elliptic curves. We define our problem again for this specific situation:

**Problem:** Let  $E$  be an elliptic curve, let  $P, Q$  on  $E$  such that  $kP = Q$ , and let  $\#E(\mathbf{F}_p) = N$ . Calculate the value of  $k$ .

We detail Pollard's  $\rho$  algorithm as follows:

(1) Partition  $G$  into three separate sets of approximately the same size.

We name these sets  $S_1, S_2, S_3$ , and we ensure that  $\mathcal{O} \notin S_2$ .

(2) Construct a random walk around  $E$  with the following function:



$$R_{i+1} = f(R_i) = \begin{cases} Q + R_i, & R_i \in S_1 \\ 2R_i, & R_i \in S_2 \\ P + R_i, & R_i \in S_3 \end{cases}.$$

(3) Let  $R_i = a_iP + b_iQ$ . Then,

$$a_{i+1} = \begin{cases} a_i, & R_i \in S_1 \\ 2a_i \bmod n, & R_i \in S_2 \\ a_i + 1, & R_i \in S_3 \end{cases}$$

and

$$b_{i+1} = \begin{cases} b_i + 1, & R_i \in S_1 \\ 2a_i \bmod n, & R_i \in S_2 \\ b_i, & R_i \in S_3 \end{cases}.$$

(4) Let  $R_0 = P$ ,  $a_0 = 1$ , and  $b_0 = 0$ .

(5) Calculate pairs of  $R_i, R_{2i}$  and record them in a list.

(6) Look for a match between  $R_m = R_{2m}$  for some value of  $m$ .

(7) When a match is found, stop. We then have  $R_m = a_mP + b_mQ$  and

$$R_{2m} = a_{2m}P + b_{2m}Q.$$

(8) Finally, compute  $k = \frac{a_{2m} - a_m}{b_m - b_{2m}} \bmod N$ .

We proceed through a modified<sup>4</sup> example from Seet [3] to demonstrate the algorithm.

*Example 10.* Consider  $E(\mathbf{F}_{47}) : y^2 = x^3 + 34x + 10$ , with  $P = (30, 26)$ ,  $Q = (35, 41)$ ,  $Q = kP$ , and  $\#E(\mathbf{F}_{47}) = N = 41$ . Solve for  $k$ .

---

<sup>4</sup> The modifications involve the numbering of the  $R_i$  and corrections of some points.

(1) First, we partition our group into three sets of similar size:

$$S_1 = \{R = (x, y) \in E(\mathbf{F}_{47}) \mid 0 \leq y < 15\},$$

$$S_2 = \{R = (x, y) \in E(\mathbf{F}_{47}) \mid 15 \leq y < 30\},$$

$$S_3 = \{R = (x, y) \in E(\mathbf{F}_{47}) \mid 30 \leq y < 47\}.$$

We note that  $|S_1| = 13$ ,  $|S_2| = 16$ , and  $|S_3| = 12$ , and thus the groups are approximately the same size.

(2) Let  $R_0 = (30, 26)$ ,  $a_0 = 1$ , and  $b_0 = 0$ .

(3) We construct the  $R_i$  as follows:

$R_i$	$a_i \pmod N$	$b_i \pmod N$	$a_iP + b_iQ$	Resulting Point
$R_0$	1	0	$1P + 0Q$	(30, 26)
$R_1$	2	0	$2P + 0Q$	(14, 9)
$R_2$	2	1	$2P + 1Q$	(34, 12)
$R_3$	2	2	$2P + 2Q$	(20, 18)
$R_4$	4	4	$4P + 4Q$	(28, 42)
$R_5$	5	4	$5P + 4Q$	(6, 17)
$R_6$	10	8	$10P + 8Q$	(30, 21)
$R_7$	20	16	$20P + 16Q$	(14, 38)
$R_8$	21	16	$21P + 16Q$	(30, 21)
$R_9$	1	32	$1P + 32Q$	(14, 38)
$R_{10}$	2	32	$2P + 32Q$	(30, 21)
$R_{11}$	4	23	$4P + 23Q$	(14, 38)
$R_{12}$	5	23	$5P + 23Q$	(30, 21)

- (4) We see that  $R_{12} = R_6 = (30, 21)$  and that  $R_{12} = R_{(2)(6)}$ , and thus we have a match. We note that  $R_6 = 10P + 8Q$ , so  $a_m = 10$  and  $b_m = 8$ , and also that  $R_{12} = 5P + 23Q$ , so  $a_{2m} = 5$  and  $b_{2m} = 23$ .
- (5) Lastly, we calculate

$$\begin{aligned}
k &= \frac{a_{2m} - a_m}{b_m - b_{2m}} \pmod{N} \\
&= \frac{5 - 10}{8 - 23} \pmod{41} \\
&= \frac{-5}{-15} \pmod{41} \\
&= \frac{1}{3} \pmod{41} \\
&= 1(3^{-1}) \pmod{41} \\
&= 14.
\end{aligned}$$

Therefore,  $Q = 14P$ , which we verify to be true.

In contrast to the Baby-Step, Giant-Step algorithm that we discussed in Section 4.1, Pollard's  $\rho$  algorithm finishes with *probabilistic measures* rather than guaranteed ones; thus, it is not certain that the algorithm will be able to solve the ECDLP in a given situation, but it may become very likely.

**4.3. Cryptographically Robust Elliptic Curves.** Despite the strength of the attacks that we have discussed, there is some methodology to choosing certain elliptic curves that are less susceptible for varying reasons. We list a few criterion from [7] here to exemplify these characteristics:

- The size of  $\#E(\mathbf{F}_p)$  used should be large. Ensuring a large group size provides more complexity since there are significantly more points that any of the attacks would proceed through. In the case of the Baby Steps, Giant Steps algorithm, it requires approximately  $\sqrt{N}$  steps, so an increase in this group size is a significant increase in the number of computations.
- There should be a large prime factor  $q$  of  $\#E(\mathbf{F}_p)$  which is comparable to the size of  $\#E(\mathbf{F}_p)$ . When choosing a starting point  $P_0$  for the algorithm, it should have order  $q$ . Ensuring these constraints causes the Pollard  $\rho$  method to take essentially the same amount of work as searching for the keys individually, and typically  $N > 2^{160}$ .
- $\#E(\mathbf{F}_p)$  should not equal  $p$  or  $p + 1$ . If it equals  $p$ , then the elliptic curve is known as *anomalous*, which renders it highly susceptible to **Smart's Attack**. If it equals  $p + 1$ , then the elliptic curve is known as *supersingular*, rendering it susceptible to other attacks. We will not discuss them here, but they are other types of cryptanalytic attacks.

## APPENDIX: APPLICATIONS AND IMPLEMENTATION

Now that we have sufficiently discussed cryptology and the foundation for its mechanisms, we can briefly highlight some direct applications to the current world.

Bitcoin is a virtual currency that has become increasingly popular in recent times. The following information about Bitcoin was gathered from [2]. Since bitcoins are not physical objects and exist solely online, they are accessed by public and private keys. The “**ECDSA**”, known as the “**Elliptic Curve Digital Signature Algorithm**”, is used to transfer ownership of bitcoins. This algorithm uses elliptic curve cryptography to generate a private key to sign each transaction and a public key to verify each transaction. By using the ECDSA, the signature of the user sending the funds can be verified for authenticity, and the user maintains the sole capability of uniquely creating the signature. In this way, rather than traditional currencies that may be backed by precious metals, Bitcoin boasts that it is backed by mathematics.

Apple’s extremely popular messaging system iMessage employs the ECDSA as well. In essence, each individual Apple device has a unique set of private and public keys, and the ECDSA works in a manner similar to that of Bitcoin. There are many other well-known platforms that have used or are currently using the ECDSA, and therefore it is essential to continue studying how to best choose cryptographically robust elliptic curves to ensure the privacy of these systems.

## REFERENCES

- [1] "Desmos Graphing Calculator". Desmos Graphing Calculator. N.p., 2015. Web. 24 Jan. 2017.
- [2] E. Rykwalder, "The Math Behind Bitcoin." CoinDesk. N.p., 19 Oct. 2014. Web. 9 Mar. 2017.
- [3] M. Seet, "Improving the Pollard-Rho Algorithm", (2007).
- [4] J. H. Silverman and J. Tate, Rational Points on Elliptic Curves, Springer-Verlag, New York, 1992.
- [5] W. Stein, Elementary Number Theory: primes, congruences, and secrets: a computational approach. Springer Science and Business Media, 2008.
- [6] L. C. Washington, Elliptic Curves: Number Theory and Cryptography. CRC Press, 2003.
- [7] E. Yin, "Curve Selection in Elliptic Curve Cryptography." (2005).
- [8] J. Zumbrägel, "LISTSERV 16.0 - NMBRTHRY Archives." LISTSERV 16.0 - NMBRTHRY Archives. N.p., 31 Jan. 2014. Web. <https://listserv.nodak.edu/cgi-bin/wa.exe?A2=NMBRTHRY;9aa2b043.1401>