Fall 2022

Cryptography Through the Lens of Group Theory

Dawson M. Shores

Winter 2022

Cryptography Through the Lens of Group Theory

Dawson M. Shores

CRYPTOGRAPHY THROUGH THE LENS OF GROUP THEORY

by

DAWSON SHORES

(Under the Direction of Paul Sobaje)

ABSTRACT

Cryptography has been around for many years, and mathematics has been around even longer. When the two subjects were combined, however, both the improvements and attacks on cryptography were prevalent. This thesis introduces and performs a comparative analysis of two versions of the ElGamal cryptosystem, both of which use the specific field of mathematics known as group theory.

INDEX WORDS: Cryptography, Group Theory, Elliptic Curves, Discrete Logarithms

2020 Mathematics Subject Classification: 11T71, 20K01

CRYPTOGRAPHY THROUGH THE LENS OF GROUP THEORY

by

DAWSON SHORES

B.S., Georgia College and State University, 2020

A Thesis Submitted to the Graduate Faculty of Georgia Southern University in Partial

Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE

STATESBORO, GEORGIA

CRYPTOGRAPHY THROUGH THE LENS OF GROUP THEORY

by

DAWSON SHORES

Major Professor:	Paul Sobaje
Committee:	Andrew Sills
	Alina Iacob
	Hua Wang

Electronic Version Approved:
December 2022

DEDICATION

First and foremost, I dedicate this paper to my fiancé, Daria. Thank you for being with me and motivating me throughout these challenging times in our life. Next I want to thank my parents, Bryan and Desiree, and my entire family for supporting me all my life. Lastly, I dedicate this paper to my three lifelong best friends: Scott, Billy, and Drew. Thanks for always being there to give me a much needed laugh, and thanks Scott for answering my countless computer science questions.

ACKNOWLEDGMENTS

I wish to acknowledge a few people. First, I want to acknowledge Dr. Sobaje for all his help and support while working on this thesis. You shared your knowledge with me, and we gained some new knowledge together. Next, I want to thank the members of my committee for their feedback and improvements on my thesis. I also want to thank the professors and students that came to my thesis defense. Lastly, I want to thank Dr. Allen at Georgia College and State University for introducing me to cryptography and helping me write the undergraduate thesis that inspired this paper.

TABLE OF CONTENTS

Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

LIST OF SYMBOLS

$\mathbb{R}$	Real Numbers
$\mathbb{C}$	Complex Numbers
$\mathbb{N}$	Natural Numbers
$\mathbb{F}$	Finite Field
$\mathbb{F}_q$	Finite Field with q elements
$\mathbb{F}_q^\times$	Group of units in $\mathbb{F}_q$
$\mathbb{Z}$	Integers
$\mathbb{Z}/n\mathbb{Z}$	Group of Integers Modulo n
$(\mathbb{Z}/n\mathbb{Z})^\times$	Group of units in $\mathbb{Z}/n\mathbb{Z}$

CHAPTER 1

INTRODUCTION

Secrecy and security are extremely important, especially in today's modern age. Cryptography is being used all around us, and has been for thousands of years. It has been used in many significant events throughout history, including war and political struggles. For a deep dive into the historical side of cryptography, read *The Code Book*, by Simon Singh [18].

In a more modern sense, cryptography is used to make your internet searches secure and keep your private information private, among other things. For example, if you open up your internet search browser and look at the certificate, chances are you will see either RSA or ECC in the certificate. These are examples of public key cryptosystems. We will define public key cryptosystems, and other important notions, in the next chapter.

Most, if not all, processes in cryptography can be broken down into mathematical operations, even if not explicitly stated as such. This includes encryption and decryption, key generation, and digital signatures. Eventually however, deeper understanding of these mathematical operations can be used to render some cryptosystems irrelevant. On the other hand, these same mathematical operations can used in the creation of new cryptosystems or the improvement of existing ones. For example, modular arithmetic can be understood and implemented without any knowledge of group theory, but once group theory is understood, the ideas used in modular arithmetic can be generalized and used in cryptography.

The main focus of this paper is to illustrate this by describing and comparing two variants of the ElGamal cryptosystem. The first variant of ElGamal we will look at is the original, that is, the process of using the group of units modulo a prime. The other variant is using points on an elliptic curve. This paper also contains introductory chapters about cryptography, group theory, and elliptic curves. Lastly, a comparison is drawn between the different attacks used to try and solve the discrete logarithm problem.

CHAPTER 2

INTRODUCTION TO CRYPTOGRAPHY

In this chapter we give an introduction to the basic terms used in and around cryptography. It is intended for the readers that are new to the field of cryptography. The chapter also provides some figures and examples to help understanding.

In cryptography, the **plaintext** is the message that we wish to send. While the plaintext can be letters, letters will usually be converted to some other form like integers, hexadecimal, binary, etc. Before the plaintext is sent, it must be encrypted. **Encryption** is the process of securing the plaintext by changing it in some way. After the plaintext is encrypted, it becomes the **ciphertext**, that is, the secured message. The person that receives the ciphertext will then decrypt it. **Decryption** is the process of undoing the encryption to bring the ciphertext back to the plaintext. The **key** is a combination of characters that are used in the encryption and decryption process. A **cipher** is a pair of algorithms where one is used for encryption and the other is used for decryption. A cipher will usually use a predetermined key. In contrast, a **cryptosystem** is a set of algorithms in which a separate algorithm is used for encryption, decryption, and key generation. A cryptosystem does not use a predetermined key since the key is created during the process of using the cryptosystem.

Example 2.1. *Our example will be a shift cipher, one of the most basic methods of encryption. Note that shift ciphers are not secure in any way, but it is being used to demonstrate the vocabulary just mentioned. We are going to let this example take place in* $\mathbb{Z}/26\mathbb{Z}$*. Suppose Alice wants to send a message to Bob. Let the message be "Thanks for reading". Assume a=0, b=1, and so on until z=25. Our plaintext will be 19,7,0,13,10,18,5,14,17,17,4,0,3,8,13,6. Let the key be 3. Alice will add 3 to each number and make sure that it is in* $\mathbb{Z}/26\mathbb{Z}$*. Then Alice has computed her ciphertext, 22,10,3,16,13,21,8,17,20,20,7,3,6,11,16,9. Which can be kept in numerical form or transformed back to letters before being sent. This would*

be wkdqnviruuhdglqj. When Bob receives the ciphertext he uses the key to decrypt the ciphertext into the plaintext. This is done by subtracting 3 from the numerical values to get 19,7,0,13,10,18,5,14,17,17,4,0,3,8,13,6. Which brings back the plaintext thanksforreading, or "Thanks for reading".

An issue with a system like this is that both Alice and Bob would have to agree upon a key ahead of time and also keep it secret. This problem was largely unaddressed until 1976 when Whitfield Diffie and Martin Hellman published their paper *New Directions in Cryptography* [2]. The two men were also helped by Ralph Merkle [18]. These men introduced public key cryptography.

Definition 1. Public key cryptography is a type of asymmetric key cryptography, meaning that there exist both a public and private key for the two parties involved. An illustration of how public key cryptography works, from [15], follows:

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

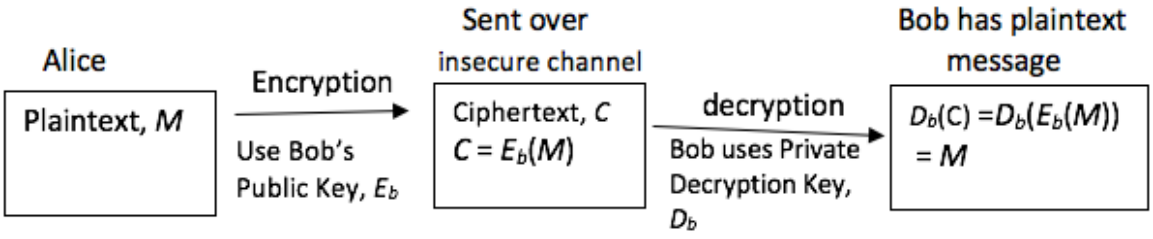

Figure 2.1: Generalized Public Key

The security of public key cryptosystems lie in the creation and usage of one-way functions.

Definition 2. A **one-way function** is a function that has the characteristic that it is easy to compute the output given the input, but hard to find the input given the output.

Some examples of one-way functions are multiplication of two large primes and modular exponentiation. For example, multiplying two large primes is relatively easy, but factoring that very large number into two prime numbers is hard.

CHAPTER 3

OVERVIEW OF GROUP THEORY

The purpose of this chapter is to give a brief introduction to the specific topics of group theory that will be used throughout this thesis. In particular, the ElGamal cryptosystem will be formulated in terms of cyclic groups. We will begin with a review of group theory and provide the necessary background information. Then we will proceed with important information regarding cyclic groups.

Definition 3. A **group** G is a set of elements with a binary operation, denoted $\cdot$, that satisfies the following properties:

1. Closure: If $a, b \in G$, then $a \cdot b \in G$.
2. Associativity: $\forall\ a, b, c \in G$, we have $(a \cdot b) \cdot c = a \cdot (b \cdot c)$.
3. Identity: $\exists\ i \in G\ \forall\ a \in G$ such that $i \cdot a = a \cdot i = a$.
4. Inverse: $\forall\ a \in G\ \exists\ a^{-1}$ such that $a^{-1} \cdot a = a \cdot a^{-1} = i$.

If a group also satisfies the commutative property, it is called an **abelian group**. The commutative property states that for all $a, b \in G$, we have $a \cdot b = b \cdot a$. One of the most recognizable abelian groups is the set of integers under addition, $(\mathbb{Z}, +)$.

Definition 4. A **field** is a set F with two operations called addition and multiplication that has the following properties:

1. Closure under both addition and multiplication.
2. F is an abelian group under addition.
3. $F \setminus \{0\}$ is an abelian group under multiplication.
4. The distributive property holds.

A field is said to be **finite**, denoted $\mathbb{F}$, if it has a finite number of elements.

Definition 5. Let G be a group and $g \in G$. In multiplication notation, we call the following the **subgroup generated by g**:

$$\langle g \rangle = \{g^a \mid a \in \mathbb{Z}\}.$$

In additive notation, this would be

$$\langle g \rangle = \{ag \mid a \in \mathbb{Z}\}.$$

Having defined the terms above, we can now define a cyclic group, highlight its important properties, and give examples.

Definition 6. Let G be a group. A group is called a **cyclic group** if $\exists\ g \in G$ such that $G = \langle g \rangle$. Then g is called a **generator** of G. Groups can have multiple generators.

The aforementioned example, $(\mathbb{Z}, +)$, is also cyclic. Indeed, one sees this since every integer can be written as either $1 + 1 + 1 + 1 + ...$ or $(-1) + (-1) + (-1) + (-1) + ...$

Example 3.1. *We can show that $(\mathbb{Z}/6\mathbb{Z}, +)$ is cyclic. Besides the obvious generator of 1, we can also show that 5 is a generator:*

$$5 = 5,\ 5{+}5 = 4, 5{+}5{+}5 = 3, 5{+}5{+}5{+}5 = 2, 5{+}5{+}5{+}5{+}5 = 1, 5{+}5{+}5{+}5{+}5{+}5 = 0$$

The **order of** G is the number of elements in the set of G, also known as the cardinality of the set. This is denoted $|G|$. When the order of the group is finite, we say G is a **finite group**. The **order of an element** $g \in G$ is the smallest positive integer a such that $g^a = i$ or $ga = i$ in multiplicative and additive notation, respectively.

Theorem 3.2. The Fundamental Theorem of Finite Abelian Groups: *Every finite abelian group can be written as the product of cyclic groups of prime powers.*

Example 3.3. *Let the order of our finite abelian groups be* $540 = 2^2 \cdot 3^3 \cdot 5$*. The following are the cyclic group possibilities for our groups:*

1. $\mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/3\mathbb{Z} \times \mathbb{Z}/3\mathbb{Z} \times \mathbb{Z}/3\mathbb{Z} \times \mathbb{Z}/5\mathbb{Z}$*;*
2. $\mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/3\mathbb{Z} \times \mathbb{Z}/9\mathbb{Z} \times \mathbb{Z}/5\mathbb{Z}$*;*
3. $\mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/27\mathbb{Z} \times \mathbb{Z}/5\mathbb{Z}$*;*
4. $\mathbb{Z}/4\mathbb{Z} \times \mathbb{Z}/3\mathbb{Z} \times \mathbb{Z}/3\mathbb{Z} \times \mathbb{Z}/3\mathbb{Z} \times \mathbb{Z}/5\mathbb{Z}$*;*
5. $\mathbb{Z}/4\mathbb{Z} \times \mathbb{Z}/3\mathbb{Z} \times \mathbb{Z}/9\mathbb{Z} \times \mathbb{Z}/5\mathbb{Z}$*;*
6. $\mathbb{Z}/4\mathbb{Z} \times \mathbb{Z}/27\mathbb{Z} \times \mathbb{Z}/5\mathbb{Z}$*.*

The following example and propositions are crucial for the rest of the paper.

Example 3.4. *Let* p *be a prime number. Then* $\mathbb{Z}/p\mathbb{Z}$*, also denoted* $\mathbb{F}_p$*, is a finite field.*

From this, one can show the following well know proposition.

Proposition 3.5. *The group of units* $(\mathbb{Z}/p\mathbb{Z})^\times$ *where* p *is a prime is always cyclic.*

It is important to note that this is not always true when p is not prime. In fact, it can be shown that $(\mathbb{Z}/8\mathbb{Z})^\times$ is not cyclic.

Lastly, we will finish with two propositions, and a consequence of these propositions.

Proposition 3.6. *Let* G *be a cyclic group and* $|G| = n$*. If* $n < \infty$*, then* $G \cong (\mathbb{Z}/n\mathbb{Z}, +)$*. If* $n = \infty$*, then* $G \cong \mathbb{Z}$*.*

Proposition 3.7. *Cyclic groups with the same order are isomorphic.*

It follows from Proposition 3.6 that $(\mathbb{Z}/p\mathbb{Z})^\times \cong (\mathbb{Z}/(p-1)\mathbb{Z}, +)$.

CHAPTER 4

ELLIPTIC CURVES

The following chapter gives the reader an introduction to elliptic curves. A lot of the more advanced topics regarding elliptic curves are glossed over, or omitted. This is because these topics are either out of the scope of this paper or not pertinent to this paper.

4.1 WHAT ARE ELLIPTIC CURVES?

Let K be a field. An elliptic curve E over K, denoted $E(K)$, is the set of points (x, y) with $x, y \in K$ that satisfy an equation of the form

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

with a, b being constants, together with a point O called the "point at infinity". The form above is if the characteristic of K is greater than 3. We call this the **Weierstrass form** or **Weierstrass equation**. There is a requirement that the discriminant $4a^3 + 27b^2 \neq 0$ to ensure that the curve has distinct roots. This condition is also known as nonsingular.

If the characteristic of $K = 2$, the equation is written as either

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

or

$$y^2 + xy = x^3 + ax^2 + b,$$

again with $a, b, c \in K$ and O.

Lastly, if $K = 3$, the equation is written in the form

$$y^2 = x^3 + ax^2 + bx + c,$$

with O.

Next we will describe the addition of points on an elliptic curve. Let $P_1 = (x_1, y_1)$ and

$P_2 = (x_2, y_2)$, be points on an elliptic curve. We define $P_3 = P_1 + P_2$. Let $P_3 = (x_3, y_3)$. The set of points on an elliptic curve form an abelian group under addition. However, addition of points on an elliptic curve is not adding the x and y coordinates. Elliptic curve point addition for the Weierstrass equation is as follows:

1. We treat O as the additive identity, that is, $P + O = P$.

2. If $P_1 \neq P_2$, then P_3 is found by drawing a line L through P_1 and P_2 to a point on the curve we will call $P_3^{'}$. Then $P_3^{'}$ is reflected over the $x-$axis to obtain P_3. This is shown in figure 4.1 from [21]. This can be done algebraically by first finding the slope m of L. Do not forget that, when dealing with modular arithmetic, "division" is actually multiplication by the inverse. Then x_3 and y_3 can be calculated using the following formulas found in [21]:

$$x_3 = m^2 - x_1 - x_2, \quad y_3 = m(x_1 - x_3) - y_1.$$

   This is under the assumption that $x_1 \neq x_2$. If $x_1 = x_2$ and $y_1 \neq y_2$, then L is vertical and the sum $= O$. This case will be excluded for the purposes as this paper.

3. If $P_1 = P_2$, then P_3 is found by drawing the tangent line L to the curve at P_1. The slope this time is found by using implicit differentiation of the Weierstrass equation. Thus we get [21]:

$$m = \frac{dy}{dx} = \frac{3(x_1)^2 + a}{2y_1}.$$

   If $y_1 = 0$, then L is vertical and O appears again. Thus, we will assume that $y_1 \neq 0$. Then, similar to above, we obtain the following formulas [21]:

$$x_3 = m^2 - 2x_1, \quad y_3 = m(x_1 - x_3) - y_1.$$

   It can be shown that point addition is also commutative and associative.

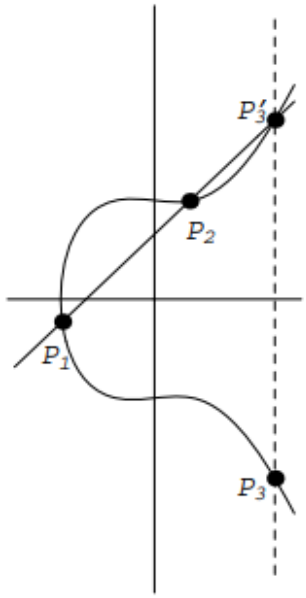

Figure 4.1: Elliptic Curve Point Addition

Example 4.1. *Let our curve be* $y^2 = x^3 + 3x + 7$ *in* $\mathbb{Z}/13\mathbb{Z}$*. We have a point* $P = (5, 2)$*. We want to calculate* $3P$*. We must first calculate* $2P$*. In order to calculate* $2P$*, we do* $P + P$*. We begin by* 2 *above.*

$$m = \frac{3 \cdot 5^2 + 3}{2 \cdot 2} = \frac{78}{4} = 78 \cdot 4^{-1} = 0 \cdot 10 = 0.$$

Then we get

$$x_3 = 0^2 - 2 \cdot 5 = -10 \equiv 3, \quad y_3 = 0(5 - 3) - 2 = -2 = 11.$$

Thus $P + P = 2P = (3, 11)$*. Now we use* 1 *to do* $P + 2P = 3P$*. First note* $P = (x_1, y_1) = (5, 2)$ *and* $2P = (x_2, y_2) = (3, 11)$*. Note that*

$$m = \frac{9}{-2} = 9 \cdot (-2)^{-1} = 9 \cdot 6 = 54 \equiv 2.$$

Next we use the formulas in 1 *to get*

$$x_3 = 2^2 - 5 - 3 = -4 = 9 \quad y_3 = 2(5 - 9) - 2 = -10 = 3.$$

Thus $3P = (9, 3)$*.*

It is useful to note that point subtraction for $P_1 \neq P_2$ is similar to point addition, however the y_2 coordinate of P_2 is negated, i.e. $P_1 - P_2 = (x_1, y_1) + (x_2, -y_2)$. When dealing with finite fields, the negative y_2 coordinate should be taken to its positive equivalent in the field. Then do point addition as in 2.

Scalar multiplication of points is not multiplying the (x, y) coordinate by the scalar n. Instead, it is adding the point to itself n times. There is another way to do this besides adding P n times. The most common way is known as "double and add". Suppose we have a point $P = (x, y)$. To calculate $151P$ we would first split 151 into $2^7 + 2^4 + 2^2 + 2^1 + 2^0$. Then we calculate

$$2P = P{+}P, \quad 2P{+}2P = 2^2P = 4P, \quad 4P{+}4P = 2^3P = 8P, \quad 8P{+}8P = 2^4P = 16P,$$

$$16P + 16P = 2^5P = 32P, \quad 32P + 32P = 2^6P = 64P, \quad 64P + 64P = 2^7P = 128P,$$

$$151P = 2^7P + 2^4P + 2^2P + 2P + P.$$

If the elliptic curve was over a finite field, the x and y value for every point would have to be kept as an element in the field.

As mentioned earlier, the points on an elliptic curve form an abelian group under addition. The requirement of closure is satisfied by how elliptic curve point addition is defined. Similarly, we defined the point at infinity as the additive identity. If $P = (x, y)$, then $-P = (x, -y)$ is the additive inverse since $P + -P = O$. There is not an easy explanation for associativity. For a proof of associativity, see [16]. Finally, the proof of commutativity is easiest to understand in a geometric sense. Elliptic curve point addition begins by drawing a line between two points. No matter the order you add the points, the same line is drawn between them, thus $P + Q = Q + P$.

CHAPTER 5

ELGAMAL

The following chapter will introduce the ElGamal cryptosystem along with the driving force behind the security of the ElGamal system.

5.1 WHAT IS ELGAMAL?

The ElGamal cryptosystem was introduced in 1985 by Taher Elgamal[1] in his paper "A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms" [3]. As the title of his paper suggests, ElGamal is a public key cryptosystem based on the discrete logarithm problem.

5.2 THE DISCRETE LOGARITHM PROBLEM

The ElGamal cryptosystem bases its security on undoing a specific one-way function on $(\mathbb{Z}/p\mathbb{Z})^{\times}$. Undoing this one-way function is called the discrete logarithm problem, (which will be shorted to DLP). The DLP can be generalized to any finite abelian group. Let G be a group with $G = \langle g \rangle$ and $\beta \in G$. Then if $\beta = g^x$, we would say that x is the discrete logarithm of β.

In $(\mathbb{Z}/p\mathbb{Z})^{\times}$, the DLP is trying to find an integer x that satisfies the congruence $\beta \equiv g^x \pmod{p}$ where p is prime and g is a primitive root modulo p. The exponent x is called the discrete logarithm of β with base g mod p and is denoted by $x = L_g(\beta)$.
For example, in $(\mathbb{Z}/479\mathbb{Z})^{\times}$ with the congruence $13^x \equiv 17 \pmod{479}$, x is the discrete logarithm of 17 with base 13 modulo 479; that is, $x = L_{13}(17)$. In this case, $x = 237$.

[1] Taher Elgamal spells his name with a lowercase g and the ElGamal cryptosystem with an uppercase G to distinguish the two [7]

5.3 How Does it Work?

Let us suppose Alice wants to send a message to Bob. We assume that the message has to be converted to an integer m_0 and that integer has been mapped to G as m. The following is the process of the ElGamal cryptosystem using a general cyclic group G:

1. Bob selects a generator g of G, a random integer r such that $2 < r < |G|$, and calculates $h = g^r \in G$. Bob sends his public key, (G, g, h), to Alice, while keeping his private key r a secret. The lower bound 2 above is chosen for security reasons.

2. Alice selects a random integer s such that $2 < s < |G|$. Then Alice calculates $\ell = g^s \in G$ and $c = mh^s \in G$. Alice sends her public key (ℓ, c) to Bob and keeps her private key s a secret.

3. Bob gets m by calculating $(c)(\ell)^{-r} = mh^s(g^s)^{-r} = m(g^{rs}(g^{-sr})) = m$.

Example 5.1. *(Note this example is pulled from [15]). Suppose Alice wants to encrypt and send the message "Math is fun!" to Bob. Let* $p = 738733242911497$*,* $g = 13$*, and* $r = 45691$*. Then observe that* $h \equiv 13^{45691}$ (*mod* 738733242911497)*; thus,* $h = 175778470844015$*. Then Bob sends* $(738733242911497, 13, 175778470844015)$ *to Alice. Then let* $s = 607512$*. Then observe that* $\ell \equiv 13^{607512}$ (*mod* 738733242911497)*; thus,* $\ell = 348425674930505$*. Alice then turns the phrase "Math is fun!" into blocks of numerical values,* m_i*, using figure 5.1. "Math"* $= 7797116104 = m_1$*, " is "* $= 3210511532 = m_2$ *(note that the space before and after "is" is included), "fun!"* $= 10211711033 = m_3$*. Then observe the following encryption of the plaintext,* m*, to ciphertext,* c*.*

$c_1 = 15303114233367 \equiv 7797116104 \cdot 175778470844015^{607512}$ (*mod* 738733242911497)

$c_2 = 110496918609746 \equiv 3210511532 \cdot 175778470844015^{607512}$ (*mod* 738733242911497)

$c_3 = 372322954090376 \equiv 10211711033 \cdot 175778470844015^{607512}$ (*mod* 738733242911497)

Then the values of ℓ, c_1, c_2, c_3 *are sent to Bob.*

Bob then takes those values and decrypts them as follows:

$$m_1 = 7797116104 \equiv 153031142333 67 \cdot 348425674930505^{-45691} \ (\textit{mod}\ 738733242911497)$$

$$m_2 = 3210511532 \equiv 110496918609746 \cdot 348425674930505^{-45691} \ (\textit{mod}\ 738733242911497)$$

$$m_3 = 10211711033 \equiv 372322954090376 \cdot 348425674930505^{-45691} \ (\textit{mod}\ 738733242911497)$$

Then using the table, the numerical values of m *are transformed back into the plaintext.*

Dec	Chr	Dec	Chr	Dec	Chr	Dec	Chr	Dec	Chr
0	NUL	26	SUB	52	4	78	N	104	h
1	SOH	27	ESC	53	5	79	O	105	i
2	STX	28	FS	54	6	80	P	106	j
3	ETX	29	GS	55	7	81	Q	107	k
4	EOT	30	RS	56	8	82	R	108	l
5	ENQ	31	US	57	9	83	S	109	m
6	ACK	32		58	:	84	T	110	n
7	BEL	33	!	59	;	85	U	111	o
8	BS	34	"	60	<	86	V	112	p
9	HT	35	#	61	=	87	W	113	q
10	LF	36	$	62	>	88	X	114	r
11	VT	37	%	63	?	89	Y	115	s
12	FF	38	&	64	@	90	Z	116	t
13	CR	39	'	65	A	91	[	117	u
14	SO	40	(	66	B	92	\	118	v
15	SI	41	)	67	C	93	]	119	w
16	DLE	42	*	68	D	94	^	120	x
17	DC1	43	+	69	E	95	_	121	y
18	DC2	44	,	70	F	96	`	122	z
19	DC3	45	-	71	G	97	a	123	{
20	DC4	46	.	72	H	98	b	124	|
21	NAK	47	/	73	I	99	c	125	}
22	SYN	48	0	74	J	100	d	126	~
23	ETB	49	1	75	K	101	e	127	DEL
24	CAN	50	2	76	L	102	f		
25	EM	51	3	77	M	103	g		

Figure 5.1: ASCII Table

5.4 ElGamal Elliptic Curve Cryptography

While generalized elliptic curve cryptography was first introduced in 1985 by Victor S. Miller [13], the first mention of using the ElGamal cryptosystem modified with elliptic curves was by Neal Koblitz in 1987 [8]. While ElGamal elliptic curve cryptography is not the most popular, it is what this paper will focus on because it is a good analog to the general group ElGamal cryptosystem. The honor of most popular elliptic curve cryptosystems belong to Elliptic Curve Digital Signature Algorithm (ECDSA) and Elliptic Curve Diffie-Hellman Key Exchanges (ECDHE) [5][20].

Before we can get into the process of ElGamal elliptic curve cryptography, we have to note that the message m must first be embedded as a the x coordinate of a point $P = (x, y)$ on the elliptic curve. This will be expanded upon later in the chapter.
Let us suppose Alice wants to send a message to Bob. The following is the process of the ElGamal elliptic curve cryptosystem using a general finite field $\mathbb{F}$:

1. An elliptic curve E over $\mathbb{F}$ is selected by Bob. He also selects a point $B \in E$ and a positive integer a. Then Bob calculates $\beta = aB$ and sends (B, β) to Alice. Note that a is Bob's private key and (B, β) is Bob's public key.

2. Alice chooses a positive integer k and uses the encoded message P to calculate $y_1 = kB$ and $y_2 = P + k\beta$. She then sends (y_1, y_2) to Bob. Note that k is Alice's private key and (y_1, y_2) is Alice's public key.

3. Lastly, Bob computes $y_2 - ay_1 = x + k\beta - akB = x + kaB - akB = P$. Then he extracts its $x-$coordinate.

Here is an example.

Example 5.2. *Bob has an elliptic curve E that is $y^2 = x^3 + 7x + 1$ over the field $\mathbb{F}_{101}$. He has a point $(28, 38) \in E(\mathbb{F}_{101})$ and a positive integer $a = 6$. Bob first calculates*

$6(28, 38) = (40, 34) = \beta$. *Bob's private key is 6, while his public key is* $((28, 38), (40, 34))$. *He sends his public key to Alice. Alice chooses a positive integer* $k = 15$ *and calculates* $15(28, 38) = (62, 10)$. *Now suppose Alice used the method described below to encode her message to the point* $(16, 13)$. *She now calculates* $(16, 13) + 15(40, 34) = (16, 13) + (1, 3) = (62, 91)$. *Note that Alice's private key is 15 and public key* $((62, 10), (62, 91))$. *Alice then sends* $((62, 10), (62, 91))$ *to Bob. Lastly, Bob computes* $(62, 91) - 6(62, 10) = (62, 91) - (1, 3) = (62, 91) + (1, 98) = (16, 13)$. *Then Bob can use the method below to take* $(16, 13)$ *and get the original message Alice sent.*

As mentioned earlier, our message must be mapped to points on our elliptic curve. We will be using Koblitz's method in [9]. However, there are other proposed methods that can be found online, including other methods proposed by Koblitz [8]. This explanation will take place in the finite field $\mathbb{F}_p$ where p is a prime. We start by choosing an integer κ such that we are satisfied with $1/2^\kappa$ being our probability that the message cannot be mapped to our curve. Usually $\kappa = 30$ suffices. Then fix an integer M such that our message m is an integer with $0 \leq m < M$. Finally, we choose a p such that $p > M\kappa$.

We now take the set of integers of the form $\{m\kappa + j\}$ with $1 \leq j < \kappa$. Then our embedded point x is defined as $x := \min\{m\kappa + j\}$ such that $\exists\ y \in \mathbb{F}_p$ that satisfies $y^2 = x^3 + ax + b$. If no such x value exists, then repeat the process with a new field.

For the recipient to take the embedded value and turn it back to the original integer value, we get $m = \lfloor (x - 1)/k \rfloor$. If we were to use F_q with $q = p^r$, we would have to first set up a one-to-one correspondence between the elements of $\{m\kappa + j\}$ and elements in $\mathbb{F}_q$. In order to explain this process in its simplest form, we will stick to $\mathbb{F}_p$ instead of $\mathbb{F}_q$.

CHAPTER 6

DLP ATTACK COMPARISON

6.1 INTRODUCTION

Despite the discrete logarithm problem being the driving force of security for multiple cryptosystems like ElGamal, Diffie-Hellman Key Exchange, and Digital Signature Algorithm, there are still attacks that exist against the DLP. The discrete logarithm problem can be attacked with two different types of algorithms, generic and non-generic. A generic algorithm is an algorithm that works for any group. A non-generic algorithm is one that only works for specific groups. The generic algorithm we will focus on in this chapter is called the "baby-step giant-step" algorithm. The non-generic algorithm is know as index calculus. One of the reasons that elliptic curve cryptography is used is that there are few attacks against the ECDLP, each working to varying success and time.

We will also discuss the implementation of baby-step giant-step to elliptic curves and explain why index calculus does not have a useful implementation for elliptic curves. Both baby-step giant-step and index calculus spawn a family of other DLP algorithms. There are other algorithms that are modifications or improvements of baby-step giant-step and index calculus. However, those will not be discussed here.

6.2 BABY-STEP GIANT-STEP

Baby-step giant-step is a generic algorithm that works for every finite cyclic group. It is also a space-time trade-off algorithm. This means that as the space, or computer storage, increases, the time it takes to compute the algorithm decreases.

Let G be a finite cyclic group with $|G| = n$. Fix g to be a generator of G and $\beta \in G$. Recall in the DLP we want to find x in $g^x = \beta$. Let α be an integer such that $\alpha = \lceil n^{\frac{1}{2}} \rceil$. The baby-step of the algorithm is to compute $g^i,$ with $i = 0, 1, ..., \alpha - 1$, and then store these

values in a table or list. The giant-step begins by calculating $\beta(g^{-\alpha})^j$ with $j = 0, 1, ..., \alpha - 1$. After each calculation, check $\beta(g^{-\alpha})^j$ against g^i in the stored baby list or table. When $\beta(g^{-\alpha})^j = g^i$, then we have now found $x = i + \alpha j$.

Now for the baby-step giant-step algorithm for an elliptic curve E over a finite field $\mathbb{F}_p$. Select a point $P \in E(\mathbb{F}_p)$ with order n, and a point Q in the subgroup generated by P. The DLP on elliptic curves is to find k such that $Q = kP$ with $0 \leq k \leq n - 1$. We say that k is the discrete logarithm of Q. The process begins by letting $\alpha = \lceil n^{\frac{1}{2}} \rceil$ and computing αP. The baby-steps portion of the algorithm is to calculate iP with $i = 0, 1, ..., \alpha - 1$ and store it as points (iP, i). Again, this can be a list or a table. The giant-step follows by calculating $Q - j(\alpha P)$ for $j = 0, 1, ..., \alpha - 1$ until there is a match between the calculated value, $Q - j(\alpha P)$, and one of the stored values, iP. Then once the matching values are found, k can be calculated by $k = i + j\alpha \pmod{n}$.

The following is an overview of the steps for the elliptic curve variation of baby-step giant-step when we are given an elliptic curve $E(\mathbb{F}_p)$, a point $P \in E$ with order n, and a point $Q \in \langle P \rangle$.

Step up: Fix an integer $\alpha = \lceil n^{\frac{1}{2}} \rceil$ and compute αP.
Baby-step: Calculate iP for $i = 0, 1, ..., \alpha - 1$ and store the value of each i and iP as a point (iP, i).
Giant-step: Calculate $Q - j(\alpha P)$ for $j = 0, 1, ..., \alpha - 1$ until $Q - j(\alpha P) = iP$, for some iP from the baby-step.
Final step: Take $i, j,$ and α and calculate $k = i + j\alpha \pmod{n}$

Table 6.1: Elliptic Curve Baby-Step Giant-Step

Example 6.1. *The following is an example of the baby-step giant-step algorithm for the elliptic curve E which is $y^2 = x^3 + 7x + 1$ over the field $\mathbb{F}_{101}$. We have the points $P = (0, 1)$, $Q = (2, 15)$, and want to find k such that $Q = kP$. The order of E is 116. Thus, $\alpha = 11$.*

Next $11P$ *is calculated. We get* $11P = (11, 55)$. *Next the baby-step,* $iP \; \forall i = 0, 1, ..., \alpha - 1$, *will be computed and put into the following list:*

$$\{[O, 0];\ [(0, 1), 1];\ [(88, 95), 2];\ [(36, 34), 3];\ [(96, 12), 4];\ [(26, 36), 5];\ [(42, 59), 6];$$

$$[(79, 39), 7];\ [(35, 87), 8];\ [(60, 68), 9];\ [(65, 19), 10]\}.$$

Then the giant-step, $Q - j(\alpha P) \; \forall \, j = 0, 1, ..., \alpha - 1$ *until* $Q - j(\alpha P) = iP$ *is found. That is,*

$$\textit{For } j = 0 : (2, 15) - 0(11, 55) = (2, 15) - O = (2, 15),$$

$$\textit{For } j = 1 : (2, 15) - 1(11, 55) = (2, 15) + (11, -55) = (2, 15) + (11, 46) = (45, 95),$$

$$\textit{For } j = 2 : (2, 15) - 2(11, 55) = (2, 15) + (1, -98) = (2, 15) + (1, 3) = (40, 34),$$

$$\textit{For } j = 3 : (2, 15) - 3(11, 55) = (2, 15) + (58, -36) = (2, 15) + (58, 65) = (36, 34).$$

Thus, we have $j = 3, i = 3$. *Then* $k = 3 + 11(3) = 36$. *So,* $Q = 36P$.

While this may seem simple on its surface, in practice much, much larger numbers are chosen.

6.3 Index Calculus

The next type of attack is index calculus. The following description of index calculus is an interpretation of [6]. We will consider the simplest case that is $\mathbb{F}_p$, with p a prime. Index calculus is a non-generic algorithm, unlike baby-step giant-step. The reason that Index calculus is not a generic algorithm will be discussed later.

First we will fix a generator g of $\mathbb{F}_p^\times$. Choose an element $\beta \in \mathbb{F}_p$. We want to find the discrete logarithm of β with respect to g i.e. $\beta = g^x \pmod p$. Note that throughout the rest of the chapter, elements or calculations being $(\mathrm{mod}\, p)$ will not be stated. Since we are in $\mathbb{F}_p$, it should be understood unless stated otherwise. Now we choose a **factor base** consisting of prime numbers and our generator, denoted $\mathcal{B} := \{g, p_1, p_2, ..., p_r\}$. The

amount of numbers in our factor base should be relatively small compared to the size of the field. Next we will find powers of the generator that factor completely into powers of the elements in our factor base and lifting it to $\mathbb{Z}$. That gives us a unique prime factorization, by the fundamental theorem of arithmetic. That is,

$$g^{k_i} \equiv p_1^{e_{i1}} \cdot p_2^{e_{i2}} \cdot ... \cdot p_r^{e_{ir}} \bmod p, \text{ for } 1 \leq i \leq j.$$

Note that j needs to be large enough to solve for a system of linear equations, at least r linearly independent relations. In practice $j = 2r$ will usually suffice. A **smooth relation** is a congruence that linearly relates logarithms of the elements of the factor base. A number is not smooth if its unique prime factorization contains primes outside of our factor base. Then we take the $\log_g$ of both sides and apply basic logarithm rules to get

$$k_i \equiv e_{i1} \log_g p_1 + e_{i2} \log_g p_2 + ... + e_{ir} \log_g p_r.$$

Next we turn this into a series of matrices to solve a system of equations for the $\log_g p$'s. That is,

$$\begin{bmatrix} e_{11} & e_{12} & ... & e_{1r} \\ e_{21} & e_{22} & ... & e_{2r} \\ \vdots & & & \\ e_{j1} & e_{j2} & ... & e_{jr} \end{bmatrix} \begin{bmatrix} \log_g p_1 \\ \log_g p_2 \\ \vdots \\ \log_g p_r \end{bmatrix} = \begin{bmatrix} k_1 \\ k_2 \\ \vdots \\ k_j \end{bmatrix} \bmod p-1.$$

Note that if we have r linearly independent relations, we should get a unique solution for the system modulo $p-1$. If we run into trouble while solving for the system of linear equations, we may need to factor $p-1$ and then use the Chinese remainder theorem. Next, we take β and multiply it by g^c. Take c to be a random integer $1 \leq c \leq p-2$. If $x = \beta g^c$ does not have a prime factorization of primes from our factor base, we choose another c and try again. When x has a prime factorization of primes from our factor base, we take $\log_g$ of both sides and perform other simplification to get,

$$x = \log_g \beta \equiv s_1 \log_g p_1 + s_2 \log_g p_2 + ... + s_r \log_g p_r - c.$$

Below is a table that outlines the steps taken in this process. We are given $\mathbb{F}_p$ with prime p, a generator g of $\mathbb{F}_p^\times$, and $\beta \in \mathbb{F}_p$. We want to find x such that $\beta = g^x$.

Step 1: Choose a factor base $\mathcal{B} := \{g, p_1, p_2, ..., p_r\}$

Step 2: Calculate $g^{k_i} \pmod p \equiv p_1^{e_{i1}} \cdot p_2^{e_{i2}} \cdot ... \cdot p_r^{e_{ir}}$, for $1 \leq i \leq j$

Step 3: Set up system of equations $k_i \equiv e_{i1} \log_g p_1 + e_{i2} \log_g p_2 + ... + e_{ir} \log_g p_r$

Step 4: Solve for $\log_g p$'s.

$$\begin{bmatrix} e_{11} & e_{12} & ... & e_{1r} \\ e_{21} & e_{22} & ... & e_{2r} \\ \vdots & & & \\ e_{j1} & e_{j2} & ... & e_{jr} \end{bmatrix} \begin{bmatrix} \log_g p_1 \\ \log_g p_2 \\ \vdots \\ \log_g p_r \end{bmatrix} = \begin{bmatrix} k_1 \\ k_2 \\ \vdots \\ k_j \end{bmatrix} \bmod p-1.$$

Step 5: Calculate $x = \beta g^c \pmod p$ for random $1 \leq c \leq p-2$. Stop when $x = p_1^{s_1} \cdot p_2^{s_2} \cdot ... \cdot p_r^{s_r}$.

Step 6: $x = \log_g \beta \equiv s_1 \log_g p_1 + s_2 \log_g p_2 + ... + s_r \log_g p_r - c \pmod p$.

Table 6.2: Index Calculus

To demonstrate the complexity of this algorithm, we will do just steps 1 and 2. Suppose we have $\mathbb{F}_{7727}$ with $g = 5$ a generator of $(\mathbb{F}_{7727})^\times$, and $1522 = \beta$. Note that 7727 is prime. We want to find x such that $1522 = 5^x \in \mathbb{F}_{7727}$. We select our factor base to be $\mathcal{B} = \{2, 3, 5, 7, 11, 13, 17\}$. Now we will raise 5 to different powers until we get numbers whose prime factors are only those of our factor base. After calculating 5^6 to 5^{60}, the only powers of 5 that were found to satisfy this were the following:

$$5^{24} = 4896 = 2^5 \cdot 3^2 \cdot 17; \quad 5^{28} = 108 = 2^2 \cdot 3^3; \quad 5^{29} = 540 = 2^2 \cdot 3^3 \cdot 5;$$

$$5^{30} = 2700 = 2^2 \cdot 3^3 \cdot 5^2; \quad 5^{45} = 5824 = 2^6 \cdot 7 \cdot 13; \quad 5^{52} = 3332 = 2^2 \cdot 7^2 \cdot 17;$$

$$5^{59} = 5324 = 2^2 \cdot 11^3.$$

However, note that $5^{28}, 5^{29}, 5^{30}$ are not linearly independent, so we only have 4 linearly independent relations. More smooth numbers need to be found to create a solvable system of linear equations. To fix this, the amount of prime numbers in the factor base needs to be reduced. If the amount of prime numbers in the factor base is reduced, then it will be easier to solve the system of equations, but it will be harder to find smooth numbers. We could also try to increase the factor base size, that way we find more smooth numbers. However, it becomes harder to solve the system of equations. To find a full example of index calculus, see [6].

The reason that this index calculus method is not generic, and thus does not have a general elliptic curve analog, is that there are no smooth numbers in elliptic curves. That is, there is no way for a power our our generator to be written as a linear combination of prime factors because there is no defined concept of a prime point. In a prime field, say $\mathbb{F}_p$, a number is smooth if it can be written as a product of powers of small primes. Since $\mathbb{F}_p \cong \mathbb{Z}/p\mathbb{Z}$, we have the homomorphism $\mathbb{Z} \to \mathbb{Z}/p\mathbb{Z}$, which allows the lifting of the elements in $\mathbb{F}_p$ to the integers so that they can be broken into prime divisors [19]. This allows for the existence of smooth numbers in $\mathbb{F}_p$, as defined earlier. This existence of smooth numbers means we can use index calculus. However, for elliptic curves, there is no such homomorphism for lifting, and thus no factor base consisting of small prime points (hence no smooth numbers). Therefore, there is no step 2. Some have suggested lifting points from $E(\mathbb{F}_p)$ to $E(\mathbb{Q})$. However, both [17] and [13] explain in depth why using $E(\mathbb{Q})$ and index calculus in general to solve ECDLP will not work.

There are special elliptic curves where index calculus does work and smooth does have a definition. The most common special curve is hyperelliptic curves, where smoothness is defined in terms of prime divisors as polynomials of certain degrees, see [4].

6.4 RSA vs ECC

As mentioned very early on in this paper, RSA is a public key cryptosystem created in 1977 by Ron Rivest, Adi Shamir, and Leonard Adleman [18], and published in their excellent paper "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems" [14]. While RSA is not the focus of this paper, it is important to note that there are multiple papers that compare the RSA to elliptic curve cryptography (shortened to ECC). There are virtually no comparisons to elliptic curve ElGamal, so elliptic curve cryptography in general is used here. The research shows that ECC outperforms RSA in most categories [10] [11] [1] [12]. This is because ECC uses smaller key lengths, and thus can perform tasks faster, as well as requiring less computer power or space.

CHAPTER 7

CONCLUSION

While cryptography and mathematics have both been around for centuries, the inclusion of mathematics to cryptography revolutionized the subject. It is important to note that number theory was most likely the first mathematical idea used in cryptography and has a lot of applications in current and past cryptosystems [15]. However, taking the systems that use number theory, like the ElGamal cryptosystem, and looking at them trough the lens of group theory has revealed new insights. This is how the elliptic curve adaptation of the ElGamal system was formed.

In this paper, we over viewed important topics in group theory and elliptic curves. Then we showed that group theory was used to improve upon the ElGamal cryptosystem. Lastly, we answered if elliptic curve cryptography was an improvement upon other number theoretic cryptosystems, and why that answer is yes. We could then ask whether or not ECC could be improved up if used with other groups. Based on the sources cited in this research, among many others, it seems that there is a high probability the answer is no. This leads us to wonder if more can be done to improve the the field of cryptography. The answer to this question seems to be pointing in the direction of computer science. And while this may be true, this research has shown that pure mathematics still holds a lot of the keys to unlocking these cryptographic mysteries.

REFERENCES

[1] M. Alimohammadi and A.A.Pouyan, "Performace Analysis of Cryptography Methods for Secure Message Exchanging in VANET", *International Journal of Scientific & Engineering Research* Vol.5 No.2 (Feb 2014).

[2] W. Diffie and M. Hellman, "New Directions in Cryptography", *IEEE Transactions on Information Theory*, VOL. IT-22. NO. 6, 644-654 (Nov 1976).

[3] T. Elgamal, "A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms", *IEEE Transactions on Information Theory*, VOL. IT-31, NO. 4, 469-472 (July 1985).

[4] P. Gaudry, "An Algorithm for Solving the Discrete Log Problem on Hyperelliptic Curves", Lecture Notes in Computer Science, vol 1807. Springer, Berlin, Heidelberg (Jan 2000).

[5] D. Holmes, *The 2016 TLS Telemetry Report*, F5 Networks (Dec 2016)

[6] J. Howell, "The Index Calculus Algorithm for Discrete Logarithms", Clemson University, 1998, https://people.clarkson.edu/∼jhowell/math/msthesis.pdf

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

[8] N. Koblitz, "Elliptic Curve Cryptosystems", Mathematics of Computation, VOL. 48. NO. 177, 203-209 (Jan 1987).

[9] N. Koblitz, *A Course in Number Theory and Cryptography*, 2nd ed. Springer-Verlag (New York) (1994).

[10] D. Mahto and D.K. Yadav, "RSA and ECC: A Comparative Analysis", (2017), *International Journal of Applied Engineering Research*. Vol.12. No.19, PP.9053-9061.

[11] D. Mahto and D.K. Yadav, "Performance Analysis of RSA and Ellitpic Curve Cryptography", (2018), International Journal of Network Security, Vol.20, No.4, PP.625-635.

[12] K. Maletsky, “RSA vs. ECC Comparison for Embedded Systems”, Microchip Technology Inc. (2020).

[13] V. Miller, “Use of Elliptic Curves in Cryptography”, in: Williams, H.C. (eds) Advances in Cryptology-CRYPTO ’85 Proceedings. CRYPTO 1985. Lecture Notes in Computer Science, VOL 218. Springer, Berlin, Heidelberg 417-426 (1986).

[14] R. L. Rivest, A. Shamir, and L. Adleman. 1978. “A Method for Obtaining Digital Signatures and Public-Key Cryptosystems”. Commun. ACM 21, 2 (Feb. 1978), 120–126. https://doi.org/10.1145/359340.359342

[15] D. Shores, “The Evolution of Cryptography Through Number Theory”, (2020), https://www.gcsu.edu/sites/default/files/documents/2021-06/shores.pdf

[16] Silverman, J.H., *The Arithmetic of Elliptic Curves*. 2nd Ed. Springer. (New York) (2016).

[17] Silverman, J.H., Suzuki, J. (1998). Elliptic Curve Discrete Logarithms and the Index Calculus. In: Ohta, K., Pei, D. (eds) Advances in Cryptology — ASIACRYPT’98. ASIACRYPT 1998. Lecture Notes in Computer Science, vol 1514. Springer, Berlin, Heidelberg.

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

[19] A. Sutherland, Lecture 11, 18.783 Elliptic Curves. 2017, math.mit.edu/classes/18.783/2017/LectureNotes11.pdf

[20] D. Warburton and S. Vinberg, *The 2021 TLS Telemetry Report*, F5 Networks, (Oct 2021)

[21] L. Washington, *Elliptic Curves: Number Theory and Cryptography*, 2nd ed. Taylor & Francis Group (Boca Raton) (2008).