PEOPLE'S DEMOCRATICREPUBLIC OF ALGERIA
MINISTERY OF HIGHER EDUCATION AND
SCIENTIFIC RESEARCH

MOHAMED BOUDIAF UNIVERSITY
-M'SILA

Faculty of Mathematics and Informatics

Department of Mathematics

Master degree in Mathematics

Domain : Mathematics and Informatics

Speciality : Mathematics

Option : Algebra and Discrete Mathematics

Title:

Elliptic Curve Cryptography

Presented by :

ABDELHAFID Abir

Defended on: June 2026.

Before the jury composed of:

President :	*GHEDBANE* Nasser	MC(A)	University of M'Sila
Supervisor :	*MIHOUBI* Douadi	Prof.	University of M'Sila
Examiner :	*HEBOUB* Lakhdar	MC(B)	University of M'Sila

Academic year: 2025/2026

الإهداء والشكر

بسم الله الرحمن الرحيم

﴿وَقُل رَّبِّ زِدْنِي عِلْمًا﴾

الحمد لله رَبّ العالمين، نحمده سبحانه وتعالى على نعمه التي لا تُحصى، ونشكرهُ على توفيقه وإعانته لنا لاتمام هذا العمل المتواضع.

أتقدم بخالص عبارات الشكر والتقدير والامتنان إلى أستاذي المشرف **الذوادي ميهوبي**، الذي لم يبخل عليّ بتوجيهاته القيّمة ونصائحه العلمية السديدة، وكان خير مرشدٍ وموجه طوال فترة إعداد هذه امذكرة. فله مني أسمى عبارات الاحترام والتقدير على ما بذلَه من جهدٍ وعطاء.

كما أتوجه بجزيل الشكر والتقدير إلى السادة أعضاء لجنة المناقشة الموقرة، الأستاذ **ناصر غضبان** والأستاذ **لخضر هبوب**، على تفضلهم بقبول مناقشة هذه المذكرة، وعلى ما سيقدمونه من ملاحظات وتوجيهات قيمة تسهم في إثراء هذا العمل وتحسينه.

كما أتقدم بالشكر إلى جميع أساتذة قسم الرياضيات بجامعة المسيلة، الذين ساهموا في تكويني العلمي، وإلى كل من ساعدني من قريب أو بعيد في إنجاز هذا العمل.

أهدي ثمرة هذا الجهد المتواضع:

إلى **والديّ** الكريمين، حفظهما الله وأدام عليهما الصحة والعافية، اللذين كانا مصدر الحب والعطاء والدعم، وغرسا في نفسي قيمة العلم والاجتهاد.

إلى **زوجي العزيز**، الذي كان سندًا لي طوال مسيرتي الدراسية، بدعمه وتشجيعه وصبره الدائم.

إلى أبنائي الأحباء **رسيم و رنيم**، اللذين كانا مصدر سعادتي وقوتي ودافعي لمواصلة العمل والنجاح.

إلى جميع أفراد عائلتي الكريمة، وإلى كل من يحمل لي المحبة والتقدير من قريب أو بعيد.

أهدي هذا العمل راجيةً من الله تعالى أن يجعله علمًا نافعًا وعملاً صالحاً.

ABDELHAFID Abir

Contents

1 Preliminary **7**

1.1 **Algebraic structures** . . . 8

1.1.1 Basic definitions of the groups , rings and fields . . . 8

1.2 **Modular Arithmetic** . . . 15

1.2.1 Modular Arithmetic . . . 15

1.2.2 Euclidean Algorithm . . . 15

1.2.3 Multiplicative Inverse . . . 16

1.2.4 Chinese Remainder Theorem . . . 17

1.2.5 Modular Exponentiation . . . 18

1.3 **Overview Cryptography** . . . 18

1.3.1 Symmetric Cryptography . . . 19

1.3.2 Asymmetric Cryptography . . . 20

2 Discrete logarithm problem **24**

2.1 **Definition of the discrete logarithm problem** . . . 24

2.2 **Algorithms for the Discrete Logarithm Problem** . . . 26

2.2.1 Brute force algorithm . . . 26

2.2.2 Baby-step Giant-step Algorithm (Shanks) . . . 27

2.2.3 Pohig-Hellman algorithm . . . 28

2.3 **Cryptographic protocols based on DLP** . . . 30

2.3.1 Diffie–Hellman key exchange . . . 30

2.3.2 ElGamal cryptosystem . . . 31

2.3.3 Digital Signature Algorithm . . . 34

3 Elliptic curve cryptography **37**

3.1 **Elliptic curve** . . . 37

3.1.1 Elliptic curve over fields . . . 38

3.1.2 Elliptic curve over finite fields . . . 39

3.1.3 Group Operations on Elliptic Curves . . . 40

3.2 **Protocols based in Elliptic Curve cryptography** 43
3.2.1 Elliptic Curve Discrete Logarithm Problem 44
3.2.2 Elliptic curve Diffie- Hellman key exchange 46
3.2.3 Elliptic curve ElGamal cryptosystem 48
3.2.4 Elliptic Curve Digital signature algorithm 50
3.2.5 ECC encryption/decryption 52
3.2.6 Comparison between RSA and ECC 54

List of Figures

1.1 Symmetric Cryptography 19
1.2 Asymmetric cryptography 22
1.3 RSA cryptosystem 23

2.1 Diffie–Hellman Key Exchange 31

3.1 Graph elliptic curve $a : y^2 = x(x+1)(x-1)$ *and* $b : y^2 = x^3 + 1$ 38
3.2 An elliptic curve $E : y^2 = x^3 + 3x + 5$ 39
3.3 Adding points of an ellipic curve 40
3.4 Doubling point of an elliptic curve 42
3.5 Inverse points of an Elliptic Curve 43
3.6 Elliptic Curve Diffie-Hellman Key Exchange 46

Introduction

Information and communication technologies have experienced remarkable development in recent decades, leading to a profound transformation of modern society and enabling the rapid exchange and processing of data on a global scale. However, this progress has been accompanied by an increasing number of security threats such as hacking, cyberattacks and identity theft, which have made it necessary to develop effective solutions to ensure information security and data protection. Cryptography has emerged as one of the fundamental scientific fields that relies on mathematical and algorithmic methods to secure information. In general, cryptography aims to achieve confidentiality, data integrity, authenticity, and non-repudiation in digital communications by transforming readable information into encrypted forms that cannot be accessed by unauthorized parties, making it a key element in protecting networks and electronic transactions.

Modern cryptographic systems are generally divided into two main categories: Symmetric cryptography and asymmetric cryptography. Symmetric Cryptography relies on a single shared secret key used for both encryption and decryption, Asymmetric Cryptography is a cryptographic system that uses a pair of keys: a public key for encryption and a private key for decryption. Although public-key cryptographic algorithms such as RSA have achieved great success, they still face certain limitations related to large key sizes and high computational cost, especially with the increasing use of resource-constrained devices and modern applications that require high speed and efficiency. In this context, elliptic curve cryptography (ECC) has emerged as a powerful and efficient alternative, as it provides a high level of security with significantly smaller key sizes compared to traditional systems, making it more suitable for modern environments such as smartphones and the Internet of Things (IoT). This type of cryptography is based on the hardness of the Elliptic Curve Discrete Logarithm Problem (ECDLP), which is considered a computationally difficult mathematical problem, thereby giving ECC a strong security foundation.

This work aims to study Elliptic Curve Cryptography (ECC) from both theoretical and practical perspectives by presenting its underlying mathematical foundations, explaining the main algorithms used in it, and highlighting its advantages compared to traditional cryptographic systems such as RSA.

The main problem addressed in this memory is: How can Elliptic Curve Cryptography (ECC) be used to achieve a high level of security while reducing key sizes and improving performance

compared to traditional systems such as RSA?
Therefore, this master's memory is divided into three main chapters as follows:

- **Preliminary** This chapter presents the necessary mathematical foundations for understanding the topic of this memory, including groups, finite fields, and some algebraic concepts used in cryptography.
- **The Discrete Logarithm Problem** (DLP) This chapter focuses on studying the discrete logarithm problem, its computational complexity, and its importance as a security foundation in public-key cryptographic systems.
- **Elliptic Curve Cryptography** (ECC) This chapter presents ECC from both theoretical and practical perspectives, including an explanation of its basic operations, the construction of cryptographic algorithm , and a comparison with traditional systems such as RSA.

Chapter 1

Preliminary

This chapter presents the fundamental mathematical concepts necessary for understanding Elliptic Curve Cryptography (ECC). It focuses on algebraic structures, modular arithmetic, and basic notions of cryptography, which form the theoretical foundation of this cryptographic system. First, algebraic structures such as groups, cyclic groups, rings and fields are introduced, as they provide the mathematical framework for cryptographic operations. Then, modular arithmetic is presented as an essential tool widely used in modern cryptography due to its efficiency and security properties. Finally, a brief overview of cryptography and its main objectives is provided in order to highlight the role of mathematics in securing information and communication systems. The concepts presented in this chapter constitute the theoretical basis for the study of elliptic curves and their cryptographic applications in the following chapters.The references used in this chapter are: [9, 18, 13, 7, 8, 4, 19]

1.1 Algebraic structures

This section introduces some fundamentals of abstract algebra, in particular the notion of groups, subgroups, rings, finite fields and cyclic groups, which are essential for understanding discrete logarithm public-key algorithms.

1.1.1 Basic definitions of the groups , rings and fields

Definition 1.1 (**Group**). *A group is a set G together with a binary operation $\circ$ that combines any two elements of G :*

$$\circ : G \times G \to G$$

$$(a, b) \to a \circ b$$

A group has the following properties.

1. *The group G is closed under the operation $\circ$. That is, for all $a, b \in G$, we have that $a \circ b = c \in G$.*
2. *The group operation is associative. That is, $a \circ (b \circ c) = (a \circ b) \circ c$ for all $a, b, c \in G$.*
3. *There exists an element $e \in G$, called the identity element, such that $a \circ e = e \circ a = a$ for all $a \in G$.*
4. *For each $a \in G$ there exists an element symmetric $b \in G$,called the inverse of a,such that $a \circ b = b \circ a = e$.*

Example 1.1.

- *$(\mathbb{Z}, +)$ is a group, i.e., the set of integers $\mathbb{Z} = \{...-2, -1, 0, 1, 2, ...\}$ together with the usual addition forms an abelian group, where $e = 0$ is the identity element and $-a$ is the inverse of an element $a \in \mathbb{Z}$.*
- *Let $\mathbb{R}$ be the set of all real numbers. Below, we verify that $\mathbb{R}$ is a group under addition. Also, we will simply assume that addition in $\mathbb{R}$ is closed and associative.*
  1. *The set $\mathbb{R}$ is closed under addition.*
  2. *Addition in $\mathbb{R}$ is associative.*
  3. *The additive identity element is $0 \in \mathbb{R}$, as $0 + a = a$ and $a + 0 = a$ for all $a \in \mathbb{R}$.*
  4. *Every $a \in \mathbb{R}$ has an additive inverse $-a \in \mathbb{R}$ such that $a + (-a) = 0$ and $(-a) + a = 0$.*

Abelian Groups:A group G is called Abelian or commutative if: $a * b = b * a$ for all $a, b \in G$. Most cryptographic groups used in public-key cryptography are Abelian group.

Definition 1.2.

- *The Additive Group $(\mathbb{Z}/n\mathbb{Z}, +)$: Let $n \geq 1$ be an integer. The set of residue classes modulo n is denoted by:*

$$\mathbb{Z}/n\mathbb{Z} = \{\overline{0}, \overline{1}, \ldots, \overline{n-1}\} = \mathbb{Z}_n$$

$$\overline{x} = x + n\mathbb{Z}$$

where each element represents an equivalence class modulo n. Two integers a and b are congruent modulo n if:
$a \equiv b \pmod{n}$ *which means:*
$n \mid (a - b)$

- *The Multiplicative Group $(\mathbb{Z}/n\mathbb{Z})^*$ modulo n is defined as:*

$$(\mathbb{Z}/n\mathbb{Z})^* = \{\overline{a} \in \mathbb{Z}/n\mathbb{Z}; gcd(a, n) = 1\}$$

with multiplication modulo n. This group plays a central role in classical public-key cryptography

Lemma 1.1. *Let $x, y \in \mathbb{Z}$ and denote by $\overline{x}, \overline{y}$ their congruence classes modulo n in $\mathbb{Z}/n\mathbb{Z}$. Then the following properties hold:*

1. *(Additive compatibility)*
$$\overline{x} + \overline{y} = \overline{x + y}.$$
2. *(Multiplicative compatibility)*
$$\overline{x} \cdot \overline{y} = \overline{xy}.$$

Proof. Let $\overline{x} = x + n\mathbb{Z}$ and $\overline{y} = y + n\mathbb{Z}$.
(1) Addition By definition of addition in $\mathbb{Z}/n\mathbb{Z}$,

$$(x + n\mathbb{Z}) + (y + n\mathbb{Z}) = (x + y) + n\mathbb{Z},$$

hence $\overline{x} + \overline{y} = \overline{x + y}$.
(2) Multiplication Similarly, by definition of multiplication,

$$(x + n\mathbb{Z})(y + n\mathbb{Z}) = xy + n\mathbb{Z},$$

hence $\overline{x} \cdot \overline{y} = \overline{xy}$. □

Example 1.2. *We verify the group properties for the set $\mathbb{Z}_7^* = \{1, 2, 3, 4, 5, 6\}$ under multiplication:*

1. *The set $\mathbb{Z}_7^*$ is closed under multiplication modulo 7. Given $a, b \in \mathbb{Z}_7^*$, their product $a \times b$ is still in $\mathbb{Z}_7$. For example, $2 \times 3 = 6 \in \mathbb{Z}_7^*$.*

2. *multiplication modulo* 7 *is associative, that is* $(a \times b) \times c = a \times (b \times c)$ *for all* $a, b, c \in \mathbb{Z}_7^*$.*After all, multiplication in* $\mathbb{Z}_7$ *is based on multiplication* $\mathbb{Z}$.*for example;*
   $(2 \times 3) \times 4 = 2 \times (3 \times 4) = 24 \equiv 3 \bmod 7$.

3. *The element* $1 \in \mathbb{Z}_7^*$ *is the multiplicative identity, because* $1 \times a = a$ *and* $a \times 1 = a$ *for all* $1 \in \mathbb{Z}_7^*$.

4. *Each element* $a \in \mathbb{Z}_7$ *has an multiplicative inverse* a^{-1} *, which is also in* $\mathbb{Z}_7$. *For example; the multiplicative inverse of* 2 *is* 4 *such that* $2 \times 4 = 8 \equiv 1 \bmod 7$.

Proposition 1.1.

- *The multiplication in* $\mathbb{Z}_n$ *is induced by the multiplication in* $\mathbb{Z}$.
- *An element* $k \in \mathbb{Z}_n$ *has a multiplicative inverse if and only if* $\gcd(k, n) = 1$.

  The set of all such invertible elements in $\mathbb{Z}_n$ *is denoted by* $U(n)$. *Then* $U(n)$ *is a group under multiplication, called the group of units of* $\mathbb{Z}_n$.
- *The order of* $U(n)$ *is given by Euler's totient function:*

$$|U(n)| = \varphi(n),$$

  where $\varphi(n)$ *is Euler's totient function, which counts the number of integers* $1 \leq k \leq n$ *such that* $\gcd(k, n) = 1$.

Remark 1.1. *That in cryptography we use both multiplicative groups, i.e., the operation "*$\circ$*" denotes multiplication, and additive groups where "*$\circ$*" denotes addition.*
The additive notation is commonly used for elliptic curve groups, as will be discussed in Chapter 3.

Definition 1.3 (**Cyclic group**). *A group* G *is called cyclic if there exists an element* $a \in G$ *such that*

$$\langle a \rangle = G.$$

Equivalently, G *is cyclic if there exists an element* $a \in G$ *whose order is equal to the order of the group:*

$$\operatorname{ord}(a) = |G|.$$

Such an element is called a generator of G

Note. If G is a cyclic group generated by a, then every element $b \in G$ can be expressed as

$$b = a^i$$

for some integer i.
Hence, the element a generates all elements of the group and is called a generator.

Example 1.3. *We want to determine whether $b = 2$ is a generator of the multiplicative group*

$$\mathbb{Z}_{11}^* = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}.$$

Since

$$|\mathbb{Z}_{11}^*| = 10,$$

we compute the powers of b modulo 11.

$b = 2$

$b^2 = 4$

$b^3 = 8$

$b^4 \equiv 5 \pmod{11}$

$b^5 \equiv 10 \pmod{11}$

$b^6 \equiv 9 \pmod{11}$

$b^7 \equiv 7 \pmod{11}$

$b^8 \equiv 3 \pmod{11}$

$b^9 \equiv 6 \pmod{11}$

$b^{10} \equiv 1 \pmod{11}$

From the last result it follows that

$ord(b) = 10 = |Z_{11}^*|$

Definition 1.4 (**Sub group**). *A subgroup H of a group G is a subset of G which is also a group with respect to the operation of G, we write in this case $H \leq G$.*

A sub group H is called trivial if it is equal to the whole group G, i.e $H = \{e\}$ or is equal to the group consisting of just the identity element $H = G$.

Theorem 1.1. *Let G be a group and let $a \in G$. Then*

$$\langle a \rangle = \{a^n : n \in \mathbb{Z}\}$$

is a subgroup of G.

This subgroup is called the cyclic subgroup generated by a.

If

$$G = \langle a \rangle,$$

then G is cyclic and a is called a generator of G.

Example 1.4. *Consider the group*

$$U(10) = \{1, 3, 7, 9\}.$$

The powers of 3 *modulo* 10 *are*

$$3^1 \equiv 3 \pmod{10},$$

$$3^2 \equiv 9 \pmod{10},$$

$$3^3 \equiv 27 \equiv 7 \pmod{10},$$

$$3^4 \equiv 21 \equiv 1 \pmod{10}.$$

Hence

$$\langle 3 \rangle = \{1, 3, 7, 9\} = U(10),$$

which shows that 3 *is a generator of* $U(10)$.

Definition 1.5. ***(Ring)*** *A set* R *together with two binary operations addition* $(+)$ *and multiplication* $(\cdot)$ *is called a ring if:*

1. $(R, +)$ *is an abelian group.*
2. *Multiplication is associative, i.e.,*

$$(ab)c = a(bc), \qquad \forall a, b, c \in R$$

3. *Multiplication is distributive over addition, i.e.,*

$$a(b + c) = ab + ac$$

and

$$(a + b)c = ac + bc, \qquad \forall a, b, c \in R$$

there exists an element $1 \in R$ *such that*

$$a \cdot 1 = 1 \cdot a = a, \qquad \forall a \in R$$

then R *is called a ring with identity.*

Example 1.5. *The set of integers modulo* 8*,*

$$\mathbb{Z}_8 = \{0, 1, 2, 3, 4, 5, 6, 7\}$$

with addition and multiplication modulo 8 *forms a ring. For example:*

$$5 + 6 \equiv 3 \pmod{8}$$

since

$$5 + 6 = 11, which \quad is \qquad 11 \equiv 3 \pmod 8$$

and

$$3 \cdot 5 = 15 \equiv 7 \pmod 8$$

Therefore, $\mathbb{Z}_8$ is a ring under addition and multiplication modulo 8.

Definition 1.6 (**Fields**). *A field $\mathbb{F}$ is a set equipped with two binary operations, addition and multiplication, satisfying the following properties:*

1. *$(\mathbb{F}, +)$ is an abelian group with the group operation $''+''$ and the neutral element 0.*
2. *$(\mathbb{F} \setminus \{0\}, \cdot)$ is an abelian group with the group operation$''X''$ and the neutral element 1.*
3. *When the two group operations are mixed, the distributivity law holds, i.e, for all $a, b, c \in \mathbb{F}$: $a(b + c) = (ab) + (ac)$, and $(b + c)a = (ba) + (ca)$*

Definition 1.7 (**Finite fields**). *A finite field is a field that contains a finite number of elements, this number is called the order of the field, and this finite field is denoted by $\mathbb{F}_q$ of q elements with $q = p^n$, where p is a prime number, and n a positive integer. Every finite field $\mathbb{F}_q$ of a prime characteristic p is an extension of the field $\mathbb{F}_p \cong \mathbb{Z}_p$ (called the prime field of characteristic p).*

Example 1.6. *We consider the finite field $\mathbb{F}_5 = \{0, 1, 2, 3, 4\}$. The tables below describe how to add and multiply any two elements, as well as the additive and multiplicative inverse of the field elements. Using these tables, we can perform all calculations in this field without using modular reduction explicitly.*

Addition

$+$	0	1	2	3	4
0	0	1	2	3	4
1	1	2	3	4	0
2	2	3	4	0	1
3	3	4	0	1	2
4	4	0	1	2	3

Multiplication

$\times$	0	1	2	3	4
0	0	0	0	0	0
1	0	1	2	3	4
2	0	2	4	1	3
3	0	3	1	4	2
4	0	4	3	2	1

Additive Inverse

$$-0 = 0$$
$$-1 = 4$$
$$-2 = 3$$
$$-3 = 2$$
$$-4 = 1$$

Multiplicative Inverse

0^{-1} does not exist

$1^{-1} = 1$

$2^{-1} = 3$

$3^{-1} = 2$

$4^{-1} = 4$

Theorem 1.2 (**Primitive element**). *Let α be a primitive element for the finite field $\mathbb{F}_q$,*

$$\mathbb{F}_q^{\times} = \{1, \alpha, \alpha^2, \ldots, \alpha^{q-2}\},$$

where $\alpha^{q-1} = 1$ Moreover, α^k is also primitive if and only if:

$$gcd(k, q-1) = 1$$

1.2 Modular Arithmetic

Almost all cryptography algorithms, both symmetric ciphers and asymmetric ciphers, are based on arithmetic with in a finite number of elements. Most number sets we are used to, such as the set of natural numbers or the set of real numbers, are infinite. In the following, we introduce modular arithmetic, which is a simple way of performing arithmetic in a finite set of integers.

1.2.1 Modular Arithmetic

Definition 1.8 (**Congruence**). *Let $a, r, m \in \mathbb{Z}$ (where $\mathbb{Z}$ is a set of all integers) and $m > 0$. We write*

$$a \equiv r \pmod{m}$$

if m divides $a - r$.
m is called the modulus and r is called the remainder.

Example 1.7. *Let $a = 42$ and $m = 9$. Then*

$$42 = 4 \times 9 + 6$$

and therefore $42 \equiv 6 \bmod 9$.

1.2.2 Euclidean Algorithm

algorithm 1.1. *The Euclidean Algorithm computes the greatest common divisor of two positive integers, say a and b. The algorithm sets r_0 to be a and r_1 to be b, and performs the following sequence of divisions:*

$$\begin{aligned}
r_0 &= q_1 r_1 + r_2 \ , \quad 0 \le r_2 < r_1 \\
r_1 &= q_2 r_2 + r_3 \ , \quad 0 \le r_3 < r_2 \\
&\cdot \ , \quad \cdot \\
&\cdot \ , \quad \cdot \\
&\cdot \ , \quad \cdot \\
r_{m-2} &= q_{m-1} r_{m-1} + r_m \ , \quad 0 \le r_m < r_{m-1} \\
r_{m-1} &= q_m r_m \ .
\end{aligned}$$

The algorithm terminates when a division yields are remainder of 0. *The last nonzero remainder,* r_m, *is the greatest common divisor of* a *and* b, *and we write* $gcd(a, b) = r_m$.

note If $gcd(a, b) = 1$, the numbers a and b are said to be relatively prime or (coprime).

Example 1.8. *We compute the greatest common divisor of 34 and 99. The euclidean algorithm proceeds as follows*

$$\begin{aligned} a &= 99, & b &= 34 \\ 99 &= 34 \cdot 2 + 31 & \Rightarrow r_1 &= 31 \\ 34 &= 31 \cdot 1 + 3 & \Rightarrow r_2 &= 3 \\ 31 &= 3 \cdot 10 + 1 & \Rightarrow r_3 &= 1 \\ 3 &= 1 \cdot 3 + 0 & \Rightarrow r_4 &= 0 \end{aligned}$$

Hence, $\gcd(99, 34) = 1$.

Theorem 1.3. ***Extended Euclidean Algorithm****:Let* a *and* b *be integers with at least one of* a, b *nonzero. There exist integers* x *and* y, *which can be found by the Extended Euclidean Algorithm, such that* $gcd(a, b) = ax + by$.

Example 1.9. *The extended euclidean algorithm can be used to express 1 as a combination of 99 and 34:* $11 \times 99 - 32 \times 34 = 1$

1.2.3 Multiplicative Inverse

Definition 1.9. *Let* $a, b \in \mathbb{Z}_n$. *We say that* a *and* b *are multiplicative inverses of each other if* $ab \equiv 1 \pmod{n}$ *i.e Together,* a *and* b *form an inverse pair. If* $a.a \equiv 1 \pmod{n}$, *then the element* a *is said to be a self-inverse.*

Example 1.10. *Now consider* $\mathbb{Z}_{15} = \{0, 1, 2, 3, ..., 12, 13, 14\}$ *(fifteen elements), where the computation is done on the* Z_{15} *clock. Let's find the elements in* $\mathbb{Z}_{15}$ *that have multiplicative inverses:*

- 0 *does not have a multiplicative inverse, because* $0.x = 0$ *for all* $x \in \mathbb{Z}_{15}$.
- *We have* $1.1 = 1$, $4.4 = 16 \equiv 1 \pmod{1}5$, $11.11 = 121 \equiv 1 \pmod{1}5$, *and* $14.14 = 196 \equiv 1 \pmod{1}5$.
  Thus, 1,4,11, *and* 14 *are self-inverses.*
- *Other inverse pairs are:* $2.8 = 16 \equiv 1 \pmod{1}5$ *and* $7.13 = 91 \equiv 1 \pmod{1}5$

The remaining elements of F_{15} *do not have multiplicative inverses. Consider* $3 \in \mathbb{Z}_{15}$, *for instance; The only multiples of* $3in\ \mathbb{Z}_{15}$ *are*$0, 3, 6, 9, 12$,*and so there is no element* $x \in \mathbb{Z}_{15}$ *such that* $3.x \equiv 1mod15$. *Therefore,the elements of* $\mathbb{Z}_{15}$ *with multiplicative inverses are* $1, 2, 4, 7, 8, 11, 13, 14$

Theorem 1.4. *Let* $n \geq 2$. *A multiplicative inverse* $a^{-1} \mod n$ *exists if and only if* $gcd(a, n) = 1$.

Example 1.11. *We noted in the previous* ***example*** *1.8 that*

$1 = 31 - 3 \times 10$

$1 = 31 - 10 \times (34 - 31) = 11 \times 31 - 10 \times 34$

$1 = 11 \times 99 - 32 \times 34$

then $-32 \times 34 \equiv 1 \mod 99$

Therefore, $34^{-1} \mod 99 \equiv -32 \mod 99 \equiv 67$.

1.2.4 Chinese Remainder Theorem

Theorem 1.5 (Chinese Remainder Theorem)**.** *Let* $n_1, n_2, \ldots, n_k$ *be integers greater than* 1 *that are pairwise coprime, i.e.,*

$$\gcd(n_i, n_j) = 1 \quad \textit{for all } i \neq j,$$

and let

$$N = n_1 n_2 \cdots n_k.$$

Then for any integers $a_1, a_2, \ldots, a_k$*, the system of congruences*

$$\begin{cases} x \equiv a_1 \mod n_1, \\ x \equiv a_2 \mod n_2, \\ \quad \vdots \\ x \equiv a_k \mod n_k \end{cases}$$

has a solution $x \in \mathbb{Z}$*. Moreover, this solution is unique modulo* N*, that is, if* x_1 *and* x_2 *are two solutions, then*

$$x_1 \equiv x_2 \mod N.$$

In particular, there exists a unique integer x *such that* $0 \leq x < N$*.*

Example 1.12. *Solve the following system of congruences:*

$$\begin{cases} x \equiv 2 \pmod 3, \\ x \equiv 3 \pmod 5, \\ x \equiv 2 \pmod 7. \end{cases}$$

First, compute

$$N = 3 \times 5 \times 7 = 105.$$

Define

$$N_1 = \frac{N}{3} = 35, \quad N_2 = \frac{N}{5} = 21, \quad N_3 = \frac{N}{7} = 15.$$

Next, we compute the inverses:

$$35^{-1} \equiv 2 \pmod 3, \quad 21^{-1} \equiv 1 \pmod 5, \quad 15^{-1} \equiv 1 \pmod 7.$$

we obtain

$$x \equiv a_1N_1y_1 + a_2N_2y_2 + a_3N_3y_3 \pmod{N}.$$

therefore,

$$x \equiv 2 \cdot 35 \cdot 2 + 3 \cdot 21 \cdot 1 + 2 \cdot 15 \cdot 1 \pmod{105}.$$

$$x \equiv 140 + 63 + 30 = 233 \equiv 23 \pmod{105}.$$

$$x \equiv 23 \pmod{105}.$$

1.2.5 Modular Exponentiation

Modular Exponentiation is considered one of the fundamental operations in number theory and modern cryptography. It is expressed as follows: $a^b \mod n$ This operation consists of computing the value of a^b and then taking the remainder upon division by n. However, in practical applications, especially in cryptography, a^b is not computed directly due to the extremely large size of the resulting value. Instead, efficient algorithms are used that rely on reducing intermediate results modulo n.

Example 1.13. *Suppose that to compute* $5^{13} \mod 23$ *in the first step convert the exponent to binary representation : 13=8+4+1 ; so* $5^{13} = 5^8 \cdot 5^4 \cdot 5^1$ *and we compute powers using repeated squaring modulo 23*

$$5^1 \equiv 5 \mod 23$$

$$5^2 = 25 \equiv 2 \mod 23$$

$$5^4 = 2^2 = 4 \mod 23$$

$$5^8 = 4^2 = 16 \mod 23$$

then we compute :

$$5^{13} \equiv 16 \cdot 4 \cdot 5 \mod 23$$

$$\textit{so } 16 \cdot 4 = 64 \equiv 18 \mod 23$$

$$18 \cdot 5 = 90 \equiv 21 \mod 23$$

finally, $5^{13} \mod 23 = 21$

1.3 Overview Cryptography

Cryptography is technique used to protect data, information and communications from the attacks by change the plaintext into incomprehensible text using symbols so that no one can understand and process it except for authorized persons. The techniques used to protect information depend on algorithms, which enable us to transmit messages in ways that make it difficult to decipher them, this is what we will present in the following sections.

Cryptographic Terminology

- **Plaintext**: the original data before encryption.
- **Ciphertext**: the encrypted data produced after applying an encryption algorithm.
- **Cipher**: the algorithm used for transforming plain text to cipher text.
- **Key**: a secret value used during the encryption and decryption process.
- **Encryption**: process of transforming plaintext into an unreadable format.
- **Decryption**: process of transforming the ciphertext back to its original form.

1.3.1 Symmetric Cryptography

In symmetric encryption, the same key k is used for both encryption and decryption. In this method, the original message is converted into an unrecognizable message. This converted message (by the sender Alice) is called the cipher text, this is done using a key and encryption algorithm. At the receiving end (the receiver is Bob), the cipher text is converted to the original message using the same key and decryption algorithm. As shown in the figure 1.1

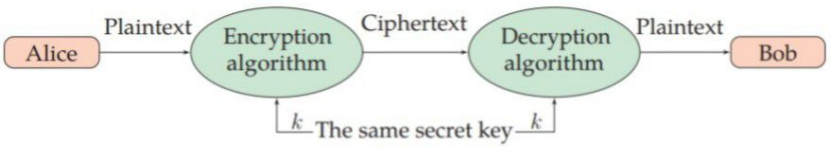


Figure 1.1: Symmetric Cryptography

Since both parties use the same key, symmetric encryption is much faster. On the other hand, the key must be available to decrypt the message. Therefore, a secure channel is required to transmit the key. In general, symmetric encryption is a straightforward method and does not require much time to complete. Some examples of symmetric encryption algorithms include:

- AES(Advanced Encryption Standard), has a key length of $128-bits, 192-bits, and 256-bits$.

- DES(Data Encryption Standard), has a key length of $56-bits$.

- $3DES$(Triple Data Encryption Standard), has a key length of 168-bits (three $56-bit\ DES$ keys.

Disadvantages of symmetric cryptography: Symmetric crypto systems have a problem of key transportation. The secret key is to be transmitted to the receiving system before the actual message is to be transmitted. All means of electronic communication is insecure as it is impossible to guarantee that no one can intercept or eavesdrop on communication channels. So the only secure way of exchanging keys would be exchanging them personally.

Example 1.14. *Let's encrypt the message ´How are you " by using AES.*
Key size in bits: $128-bits$.
Secret key: $azertyuiopqsdfgh$
Output text format: hexadecimal.
AES encrypted output: $"698F154A3C9F833C1E01C4A1BFDDED63\ "$
Let's decrypt the message
$"177C1A81C2D94F9D6BFC1DBAFB3F99371C4D6BAE7E2CB790FE4A95249DCD6FC0\ "$
by using AES.
Input text format: hexadecimal.
Key size in bits: $128-bits$.
Secret key: $azertyuiopqsdfgh$
AESdecrypted output: $"Iamfinethankyou\ "$.

1.3.2 Asymmetric Cryptography

Different from symmetric ciphers, asymmetric ciphers are also called **public-key cryptography**. This concept was introduced in 1976 by Diffie and Hellman in their article entitled *New Directions in Cryptography* For each user, we have:

$$K = (K_{priv}, K_{pub})$$

where:

$$K_{priv} : \text{private key}$$

$$K_{pub} : \text{public key}$$

For every public key K_{pub}, there exists an encryption function:

$$E_{K_{pub}} : M \to C$$

For every private key K_{priv}, there exists a decryption function:

$$D_{K_{priv}} : C \to M$$

with the property:

$$D_{K_{priv}}\left(E_{K_{pub}}(M)\right) = M$$

where:

- M : plaintext message
- C : ciphertext

The public key is shared openly before communication, while the private key must remain secret. Only the owner of the private key can decrypt the encrypted message. Asymmetric cryptography is a method of encrypting data also known as public key cryptography with two different keys , one of the keys available for anyone to use called the public key and the other key is known as the private key. Together, they are used to encrypt and decrypt data. If you encrypt a data using a person's public key, they can only decrypt it using their matching private key , which ensures a high level of security and protects the information.
If Alice (sender) uses public key cryptography (PKC), she encrypts the message using Bob's (receiver) public key and Bob will be able to decrypt it using his private key, asymmetric encryption therefore provides a higher level of security because even if someone intercepts their messages and finds Bob's public key, they will not be able to decrypt the message or understand its content.
The process of asymmetric encryption between Alice and Bob is shown in the following figure1.2

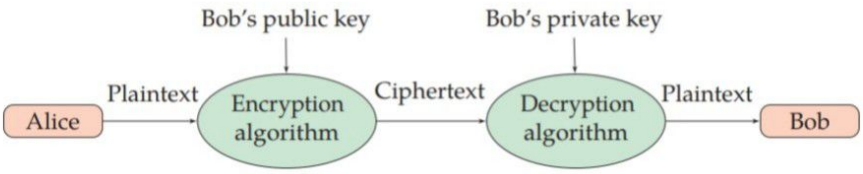


Figure 1.2: Asymmetric cryptography

Asymmetric Encryption Algorithms include:

- Rivest-Shamir-Adleman (RSA).
- Digital Signature Algorithm (DSA).
- Diffie-Hellman algorithm ($D - H$).
- Elliptic Curve Cryptography (ECC).

Disadvantages of asymmetric cryptography: a disadvantage of using public-key cryptography for encryption is speed: there are popular symmetric key encryption methods which are significantly faster than any currently available public-key encryption method.

RSA cryptosystem: The RSA cryptosystem, also known as the Rivest–Shamir–Adleman algorithm, is one of the most widely used asymmetric cryptographic schemes.

RSA encryption and decryption are performed in the ring $\mathbb{Z}_n$, and modular arithmetic plays a central role in its operations..

Let $n = pq$, where p and q are primes. Let $P = C = \mathbb{Z}_n$, and define

$$K = \{(n, p, q, a, b) : ab \equiv 1 \pmod{\phi(n)}\}$$

$$\phi(n) = (p-1)(q-1)$$

For $K = (n, p, q, a, b)$, define:

$$E_k(x) = x^b \bmod n$$

and

$$D_k(y) = y^a \bmod n$$

$(x, y) \in \mathbb{Z}_n^*$, the values n and b comprise the public key, and the values p, q, a form the private key.

Figure 1.3: RSA cryptosystem

Example 1.15. *Let Alice and Bob operate the RSA cryptosystem.*

$$p = 5, \quad q = 11, \quad n = 55, \quad \varphi(n) = 40$$

Alice selects:

$$e = 3, \quad \gcd(e, \varphi(n)) = 1$$

and computes:

$$d \equiv e^{-1} \pmod{\varphi(n)} \Rightarrow d = 27$$

Thus:

$$(n, e) = (55, 3), \qquad (n, d) = (55, 27)$$

Let $m \in \mathbb{Z}_n$ with $m = 7$. Encryption:

$$c \equiv m^e \pmod{n} \equiv 7^3 \pmod{55} = 13$$

Decryption:

$$m \equiv c^d \pmod{n} \equiv 13^{27} \pmod{55} = 7$$

Chapter 2

Discrete logarithm problem

The security of the RSA cryptosystem is fundamentally based on the computational difficulty of factoring large integers. Another important hard problem in number theory is the Discrete Logarithm Problem (DLP). Discrete logarithms problem are fundamental to several public-key cryptographic algorithms, including the Diffie-Hellman key exchange, ElGamal encryption, and the Digital Signature Algorithm (DSA).The references used in this chapter are:[9, 17, 18, 14, 15, 5, 6, 22]

2.1 Definition of the discrete logarithm problem

Definition 2.1. ***(DLP in*** $\mathbb{Z}_p^*$*)* *Given is the finite cyclic group* $\mathbb{Z}_p^*$ *of order* $p-1$ *and a primitive element* $\alpha \in \mathbb{Z}_p^*$ *and another element* $\beta \in \mathbb{Z}_p^*$*. The DLP is the problem of determining the integer*

$$1 \leq x \leq p-1 \text{ such that: } \alpha^x \equiv \beta \ (mod\ p)$$

so that x *is called the discrete logarithm of* β *to the base* α*, and we can formally write:*

$$x = log_\alpha(\beta) \ (mod\ p).$$

Example 2.1. *Consider the finite cyclic group* $\mathbb{Z}_{11}^*$*, in which* $\alpha = 2$ *is a primitive element for* $\beta = 9$*.*

The discrete logarithm problem is to find the positive integer x *such that:* $2^x = 9 \ (mod\ 11)$*."by*

using a brute force attack"

$$2^1 \equiv 2 \ (mod\ 11)$$
$$2^2 \equiv 4 \ (mod\ 11)$$
$$2^3 \equiv 8 \ (mod\ 11)$$
$$2^4 \equiv 5 \ (mod\ 11)$$
$$2^5 \equiv 10 \ (mod\ 11)$$
$$2^6 \equiv 9 \ (mod\ 11)$$
$$2^7 \equiv 7 \ (mod\ 11)$$
$$2^8 \equiv 3 \ (mod\ 11)$$
$$2^9 \equiv 6 \ (mod\ 11)$$
$$2^{10} \equiv 1 \ (mod\ 11)$$

Thus:

$$x = 6$$

Since the order of 2 *in* $\mathbb{Z}_{11}^*$ *is* 10*, the complete set of solutions is*

$$x \equiv 6 \pmod{10}$$

or equivalently

$$x = 6 + 10k, \qquad k \in \mathbb{Z}.$$

Definition 2.2. ***(The Generalized*** DLP***)***

The Generalized Discrete Logarithm Problem Given a finite cyclic group G *with the group operation* $\circ$ *and cardinality* n*. We consider a primitive element* $\alpha \in G$ *and another element* $\beta \in G$*. The discrete logarithm problem is finding the integer* x*, where* $1 \leq x \leq n$*, such that:*

$$\beta = \underbrace{\alpha \circ \alpha \circ ... \circ \alpha}_{x times}$$

then:

- *if* $\circ$ *is* ***multiplication****, then:*
$$\beta = \alpha^x$$
- *if* $\circ$ *is* ***addition****, then:*
$$\beta = x \cdot \alpha$$

Example 2.2. *Consider the additive group of integers modulo a prime* $p = 11$*,* $G = (\mathbb{Z}_{11}, +)$ *is a finite cyclic group with the primitive element* $\alpha = 2$*.*

x	1	2	3	4	5	6	7	8	9	10	11
$x \cdot \alpha$	2	4	6	8	10	1	3	5	7	9	0

let's solve the DLP for the element $\beta = 5$, i.e:

$$x \cdot 2 = \underbrace{2 + 2 + ... + 2}_{x times} \equiv 5 \ (mod\ 11)$$

in order to solve for x, we simply have to invert the primitive element α. $x = 2^{-1} \cdot 5 \ (mod\ 11)$ we can compute $2^{-1} \equiv 6 \ (mod\ 11)$ (using the extended euclidean algorithm).
Hence,

$$x \equiv 6 \times 5 \ (mod\ 11)$$

$$x \equiv 8 \ (mod\ 11)$$

Remark 2.1. *The problem of calculating the discrete logarithm is generally a difficult problem (more or less depending on the group G). For suitable group such as $(Z/nZ)^*$ this is considered to be a hard problem. We see that for the additive group of $\mathbb{Z}/n\mathbb{Z}$ the DLP is easy to solve. A much better choice are the multiplicative groups $(\mathbb{Z}/n\mathbb{Z})^*$, which are cyclic if and only if $n = 2, 4, p^r \quad or \quad 2p^r$, where $p > 2$ is a prime and $r > 1$.*

2.2 Algorithms for the Discrete Logarithm Problem

Let $(G, \cdot)$is a multiplicative group and $\alpha \in G$ has order n. hence the discrete logarithm problem can then be stated as follows: given $\beta \in < \alpha >$, find the unique exponent x, $0 \leq x \leq n-1$, such that $\alpha^x = \beta$.

2.2.1 Brute force algorithm

A brute-force attack is one of the simplest and most generic attacks in cryptography. It consists of exhaustively searching all possible values of a secret parameter until the correct one is found. this type of attack does not exploit any structural weakness in cryptographic algorithm; instead

,it relies purely on computational power:

$$\alpha^1 \stackrel{?}{=} \beta$$
$$\alpha^2 \stackrel{?}{=} \beta$$
$$\alpha^3 \stackrel{?}{=} \beta$$
$$.$$
$$.$$
$$.$$
$$\alpha^x \stackrel{?}{=} \beta$$

see **example** 2.1

2.2.2 Baby-step Giant-step Algorithm (Shanks)

is a time memory trad-off method, which reduces the time of a brute force. the idea is based on rewriting the DL $x = log_\alpha \beta$ in a two-digit representation:

$$x = mj + i \quad \text{for} \quad 0 \leq i, j \leq m - 1$$

so that m= $\left\lfloor \sqrt{|G|} \right\rfloor$, we can now write the discrete logarithm as

$$\beta = \alpha^x = \alpha^{mj+i}$$
$$i.e \quad \alpha^{mj} = \beta\alpha^{-i} = K$$

observe that: $(j; k) \in L_1$ and $(i; k) \in L_2$
as desired. conversely, for any $\beta \in G$, we have that $0 \leq log_\alpha \beta \leq n - 1$, if we devide $log_\alpha \beta$ by the integer m, then we can express $log_\alpha \beta$ in the form : $log_\alpha \beta = mj + i$.

algorithm 2.1. *SHANKS* $[G, n, \alpha, \beta]$

- $m = \lceil \sqrt{n} \rceil$
- *for* $0 \leq j \leq m - 1$ *, compute* α^{mj}
- *Sort the* m *ordered pairs* (j, α^{mj}) *with respect to their second coordinates, obtaining a list* L_1
- *for* $0 \leq i \leq m - 1$ *, compute* $\beta\alpha^{-i}$
- *Sort the* m *ordered pairs* $(i, \beta\alpha^{-i})$ *with respect to their second coordinates, obtaining a list* L_2
- *Find a pair* $(j, k) \in L_1$ *and* $(i, k) \in L_2$ *(i.e, find tow pair having identical scond coordinates)*
- $log_\alpha \beta = mj + i \bmod n$

Example 2.3. *Suppose we which to find* $log_2 100$ *in* $(\mathbb{Z}^*_{131}, \cdot)$*, note that 131 is prime and 2 is a primitive element in* $\mathbb{Z}^*_{131}$*, so we have* $\alpha = 2$ *,* $n = 130$*,* $\beta = 100$*, and* $m = \lceil \sqrt{130} \rceil = 12$*, then* $\alpha^{12} \bmod 131 = 35$*.*

First, we compute the order pair $(j, 35^j) \bmod 131$ *, for* $0 \leq j \leq 11$

we obtain the list:

$$(0,1), (1,35), (2;46)$$
$$(3,38), (4,20), (5,45)$$
$$(6,3), \textbf{\textit{(7,105)}}, (8,7)$$
$$(9,114), (10,60), (11,4)$$

which is then sorted to produce l_1

The second list contains the ordered pairs $(i, 100 \times (2^{-1})^i \bmod 131)$ *, for* $0 \leq i \leq 11$

(i.e, $(i, 100 \times 66^i \bmod 131)$*, it is as follows :*

$$(0,100), (1,50), (2,25)$$
$$(3,78), (4,39), (5,85)$$
$$(6,108), (7,54), (8,27)$$
$$(9,79), \textbf{\textit{(10, 105)}}, (11,118)$$

after sorting this list we get L_2 *.*

if we proceed simultaneously through the tow sorted lists,we find that $(7, 105)$ *in* L_1 *and*$(10, 105)$ *in* L_2*. hence, we can compute*

$log_2 100 = (12 \times 7 + 10) \bmod 131 = 94$

as a check , it can be verified that $2^{94} = 100 \bmod 131$*.*

2.2.3 Pohig-Hellman algorithm

This algorithm is used to solve the DLP in finite cyclic group, it is particularly efficient when the order of the group can be factored into small prime powers , this algorithm is based on the Chinese remainder theorem.

Suppose that $n = \prod_{i=1}^{t} p_i^{e_i}$, where the p_i's are distinct primes.Since $x = \log_\alpha \beta$ is determined modulo n, it is sufficient to compute $x \pmod{p_i^{e_i}}$ for each factor $p_i^{e_i}$, $1 \leq i \leq t$. then we can compute $x \bmod n$ by Chinese remainder theorem .

algorithm 2.2. *Pohlig-Hellman*

- *Factorize* $n = \prod_{i=1}^{k} p_i^{e_i}$
- *each* $i = 1$ *to* t
  Compute $\alpha_i = \alpha^{n/p_i^{e_i}}$

Compute $\beta_i = \beta^{n/p_i^{e_i}}$

- *Solve* $\alpha_i^{x_i} = \beta_i \pmod{p_i^{e_i}}$
  so that : $x = x_0 + x_1 p_1 + x_2 p_2^2 + ...$

- *Use the Chinese Remainder Theorem to find* x *such that:*

$$x \equiv x_i \pmod{p_i^{e_i}} \quad \textit{for all } i$$

Example 2.4. *Solving* $2^x \equiv 18 \pmod{29}$ *using Pohlig-Hellman We solve the discrete logarithm:*

$$2^x \equiv 18 \pmod{29}$$

Since 29 *is prime, we work in the multiplicative group:*

$$|(\mathbb{Z}/29\mathbb{Z})^\times| = 28 = 4 \times 7$$

We compute:

$$x \bmod 4 \quad \textit{and} \quad x \bmod 7$$

Computation modulo 7 Let:

$$e_7 = \frac{28}{7} = 4$$

Compute:

$$\alpha_7 = 2^4 \equiv 16 \pmod{29}$$

$$\beta_7 = 18^4 \pmod{29}$$

Now:

$$18^2 = 324 \equiv 5 \pmod{29}$$

$$18^4 = 5^2 = 25 \pmod{29}$$

So we solve:

$$16^{x_7} \equiv 25 \pmod{29}$$

Checking powers:

$$16^4 \equiv 25 \pmod{29}$$

Hence:

$$x \equiv 4 \pmod{7}$$

Computation modulo 4 Let:

$$e_4 = \frac{28}{4} = 7$$

Compute:

$$\alpha_4 = 2^7 \equiv 12 \pmod{29}$$

Now compute:

$$18^7 \equiv 17 \pmod{29}$$

So:

$$12^{x_4} \equiv 17 \pmod{29}$$

Checking powers:

$$12^3 \equiv 17 \pmod{29}$$

Hence:

$$x \equiv 3 \pmod{4}$$

Chinese Remainder Theorem We solve:

$$x \equiv 3 \pmod{4}$$

$$x \equiv 4 \pmod{7}$$

Testing values gives:

$$x = 11$$

$$\boxed{x = 11}$$

2.3 Cryptographic protocols based on DLP

2.3.1 Diffie–Hellman key exchange

The Diffie-Hellman key exchange (DHKE), proposed by Whitfield Diffie and Martin Hellman in 1976, was the first asymmetric scheme published in the open literature. It enables two parties to derive a common secret key by communicating over an insecure channel. The DHKE is a very impressive application of the discrete logarithm problem that we will study in the subsequent sections. This fundamental key agreement technique is implemented in many open and commercial cryptographic protocols like Secure Shell (SSH), Transport Layer Security (TLS), and Internet Protocol Security (IPSec). The basic idea behind the DHKE is that exponentiation in $\mathbb{Z}_p^*$ works as follows:

$$k = (\alpha^x)^y \equiv (\alpha^y)^x \mod p$$

Let us now consider how the Diffie–Hellman key exchange protocol over $\mathbb{Z}_p^*$ works. In this protocol, we have two parties, Alice and Bob, who would like to establish a shared secret key using the Diffie–Hellman protocol. **key exchange protocol**

1. Alice and Bob publicly agree on a prime modulus number p, and a generator α of the multiplicative group $\mathbb{Z}_p^*$ and $1 < \alpha < p$.

2. Verify that p and α are co-primes , i.e. $gcd(p, \alpha) = 1$.

3. Alice selects a random private key x with $1 \leq x \leq p-2$, then calculates a public key $\alpha^x \bmod p$

4. Bob selects a random private key y with $1 \leq y \leq p-2$, then calculates a public key $\alpha^y \bmod p$

5. Alice sends Bob the public key Y_a, and Bob sends Alice the public key Y_b.

6. Alice computes $K_1 \equiv (\alpha^y)^x \bmod p$.

7. Bob computes $K_2 \equiv (\alpha^x)^y \bmod p$.

8. Verify that $K = K_1 = K_2$, where K_1 and K_2 are the calculated shared secret key.

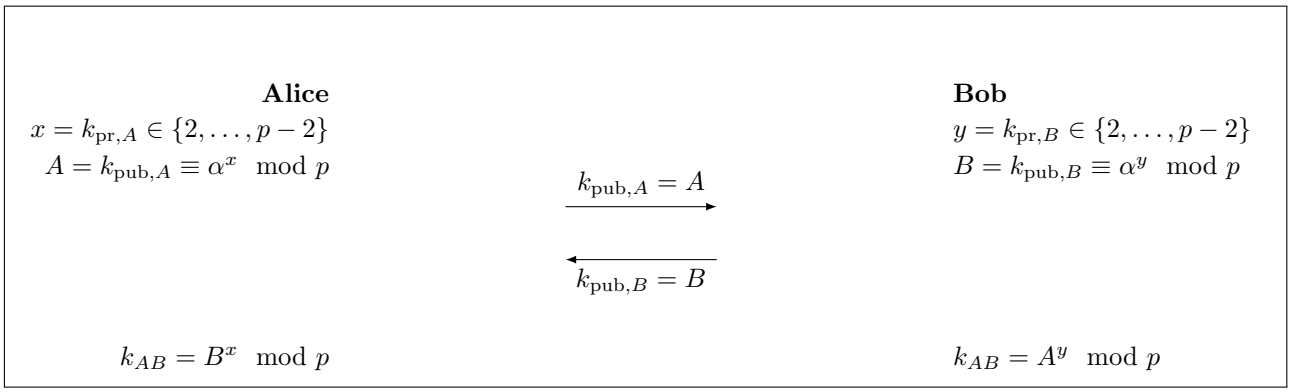
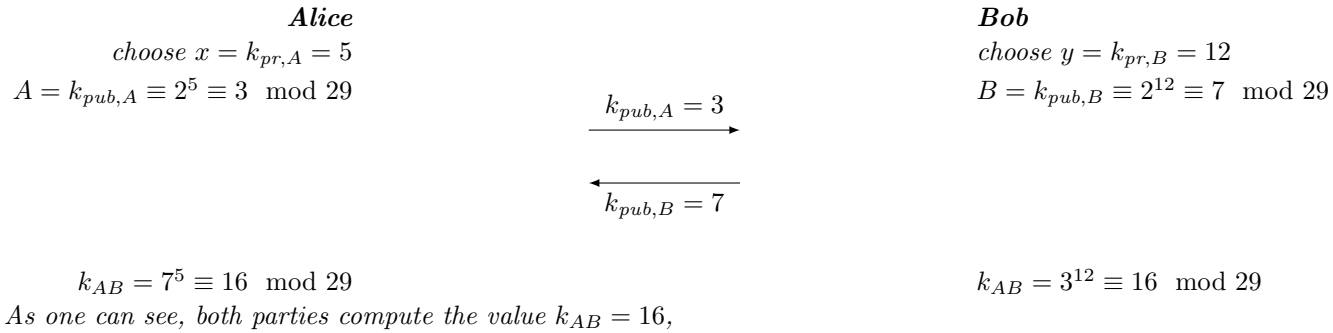


Figure 2.1: Diffie–Hellman Key Exchange

Example 2.5. *Suppose Alice and Bob are two users, to exchange the secret key between them. Let* $p = 29$ *and let* $\alpha = 2$ *be a primitive root modulo* 29*. the protocol proceeds as follows:*



As one can see, both parties compute the value $k_{AB} = 16$*,*

2.3.2 ElGamal cryptosystem

In 1985; ElGamal introduced a public key cryptosystem can be based on the discrete log problem in $(\mathbb{Z}_p^*; .)$. The system uses a public key for encryption and a private key for decryption , allowing secure exchange of information over insecure networks in an efficient and reliable way.

key Generation

- choose a large prime number p
- choose a generator α of the multiplicative group $\mathbb{Z}_p^*$
- choose a private key x (random integer)
- compute public key:
  $\beta = \alpha^x \bmod p$
- public key:(p, α, β)
- private key: x

ElGamal Encryption

to encrypt a message m:

- choose a random number k
- compute:
$$\begin{cases} c_1 = \alpha^k \bmod p \\ c_2 = m \cdot \beta^k \bmod p \end{cases}$$
- ciphertext (c_1, c_2)

ElGamal Decryption

using private key x

- compute:
  $m = c_2 \cdot (c_1^x)^{-1} \bmod p$
  $c_1^x = (\alpha^k)^x$

difficulty of computing x from $\beta = \alpha^x \bmod p$. this is exactly the discrete logarithm problem , which is hard for large numbers.

Example 2.6. *Suppose* $p = 131$ *and* $\alpha = 2$, α *is primitive element modulo p. Let* $x = 94$, *so* $\beta = \alpha^x \mod p = 2^{94} \mod 131 = 100$ *Now, suppose that Alice wishes to send the message* $m = 112$ *to Bob. so she is choose a random integer* $k = 13$ *and Compute* c_1

$$c_1 = \alpha^k \mod p = 2^{13} \mod 131$$

$$2^{13} = 8192$$

$$c_1 = 8192 \mod 131 = 70$$

and Compute $\beta^k \mod p$

$$\beta^k = 100^{13} \mod 131$$

Using modular exponentiation:

$$100^2 \equiv 44 \mod 131$$

$$100^4 \equiv 102 \mod 131$$

$$100^8 \equiv 55 \mod 131$$

Thus:

$$100^{13} = 100^8 \cdot 100^4 \cdot 100$$

$$= 55 \cdot 102 \cdot 100 \mod 131$$

$$55 \cdot 102 \equiv 108 \mod 131$$

$$108 \cdot 100 \equiv 58 \mod 131$$

So:

$$\beta^k \equiv 58 \mod 131$$

Compute c_2

$$c_2 = m \cdot \beta^k \mod p$$

$$c_2 = 112 \cdot 58 \mod 131$$

$$112 \cdot 58 = 6496$$

$$c_2 = 6496 \mod 131 = 77$$

when Bob receives the Cipher text

$$(c_1, c_2) = (70, 77)$$

and he compute

$$m = c_2 \cdot (c_1^x)^{-1} \mod p$$

$$m = 77 \cdot (70^{94})^{-1} \mod 131$$

$$77 \cdot 61 = 4697$$

$$m = 4697 \mod 131 = 112$$

$$m = 112$$

which was the plaintext that Alice encrypted.

2.3.3 Digital Signature Algorithm

The Digital Signature Algorithm (DSA) was proposed in 1991 by the U.S. National Institute of Standards and Technology (NIST), NIST chose to base their signature scheme on the discrete logarithm problem in a prime finite field $\mathbb{F}_p$, which are difficult to compute.
Digital signature is an example of asymmetric key cryptography which uses three different algorithms to complete the process:

- **key generation algorithm**: is key generation algorithm which generates private key and a corresponding public key.
- **signature generation algorithm**: is signing algorithm which selects sending both of the message and the signature using a private key generated in the first step and hash function.
- **signature verification algorithm**: is signature verifying algorithm which verifies the authenticity of sending message using a public key.

Key generation Algorithm
The keys for Digital Signature Algorithm are computed as follows:

- Generate a prime number p, where $2^{L-1} < p < 2^L$ for $512 \leq L \leq 1024$ and L a multiple of 64 i.e, $L \in \{512, 576, 640, 704, 768, 832, 896, 960, 1024\}$.
- Find a prime divisor q of $(p-1)$, with $2^{159} < q < 2^{160}$.
- Choose an integer α, $(1 < \alpha < p)$, satisfying the two conditions $\alpha^q \bmod p = 1$ and $\alpha \equiv h^{(p-1)/q} mod p$, where h is any integer with $1 < h < p-1$ such that $h^{(p-1)/q} \bmod p > 1$.

The algorithm parameters (p, q, g) may be shared between different users of the system. The second phase computes private and public keys for a single user:

1. Choose a random integer x, such that $0 < x < q$
2. Compute $\beta \equiv \alpha^x mod p$.
3. The keys are now:

$$\begin{cases} the \quad public \quad key \quad is \quad k_{pub} = (p, q, \alpha, \beta). \\ the \quad private \quad key \quad is \quad K_{pr} = x \end{cases}$$

signature generation algorithm
Let m be a message and $h(m)$ be a hash value of m. The signature generation process is as follows:

1. Choose a random per-message secret number k, where $0 < k < q$.
2. Compute $r \equiv (\alpha^k \bmod p) \bmod q$.

3. Compute $s \equiv [k^{-1}(h(m) + xr)] \bmod q$.

4. If $r = 0$ or $s = 0$ then go to step 1.

5. Sends the message m and the signature (r, s) to receiver.

Signature verification algorithm

The signature verification process is as follows:

1. Verify that: $0 < r < q$ and $0 < s < q$.

2. Compute $w \equiv s^{-1} \bmod q$.

3. Compute $u_1 \equiv [h(m)w] \bmod q$.

4. Compute $u_2 \equiv rw \bmod q$.

5. Compute $v \equiv [(\alpha^{u_1}\beta^{u_2}) \bmod p] \bmod q$.

6. The verification: v

$$\begin{cases} \equiv r \bmod q \Rightarrow & valid \quad signature. \\ \not\equiv r \bmod q \Rightarrow & invalid \quad signature \quad and \quad rejects \quad the \quad signature. \end{cases}$$

Proof (Correctness of the DSA algorithm). To prove the DSA algorithm requires showing that a signature (r, s) satisfies the verification condition $v \equiv r \bmod q$. The signer computes:

$$s \equiv [k^{-1}(h(m) + xr)] \bmod q$$

Thus

$$k \equiv h(m)s^{-1} + xrs^{-1} \bmod q$$

$$\equiv h(m)w + xrw \bmod q$$

Since α has order $q \bmod p$ we have:

$$\alpha^k \equiv \alpha^{h(m)w}\alpha^{xrw} \bmod q$$

$$\equiv \alpha^{h(m)w}\beta^{rw} \bmod q$$

$$\equiv \alpha^{u_1}\beta^{u_2} \bmod q$$

Finally, the correctness of DSA follows from:

$$r \equiv (\alpha^k \bmod p) \bmod q$$

$$\equiv (\alpha^{u_1}\beta^{u_2} \bmod p) \bmod q$$

$$\equiv v \bmod q.$$

Example 2.7. *Alice and Bob are two users, Alice wants to send a message m to Bob which is to be signed with the DSA. Suppose the hash value of m is h(m) = 30. Then the signature and verification process is as follows:*

Alice

- *Key generation:*

$$\begin{cases} Choose : p = 61, q = 5, g = 9 \\ Choose \quad private \quad key : x = 4 \\ Compute : \beta \equiv \alpha^x \bmod 61 = 34 \\ Public \quad key : (p, q, \alpha, \beta) = (61, 5, 9, 34) \end{cases}$$

- *Sign:*

$$\begin{cases} Compute \quad hash \quad of \quad message : h(m) = 30 \\ Choose \quad ephemeral \quad key : k = 3 \\ Compute : r \equiv (9^3 \bmod 61) \bmod 5 = 3 \\ Compute : s \equiv [2 \cdot (30 + 4 \cdot 3)] \bmod 5 = 4 \\ SendstoBob : (m, (r, s)) = (m, (3, 4)) \end{cases}$$

BOB

- *Verify:*

$$\begin{cases} Compute : w = 4^{-1} \equiv 4 bmod 5 \\ Compute : u_1 = 4 \cdot 30 \equiv 0 \bmod 5 \\ Compute : u_2 = 3 \cdot 4 \equiv 2 mod 5 \\ Verification : v \equiv [90 \cdot 342 \bmod 61] \bmod 5 = 3 \\ v \equiv r \bmod 5 \Rightarrow \quad valid \quad signature \end{cases}$$

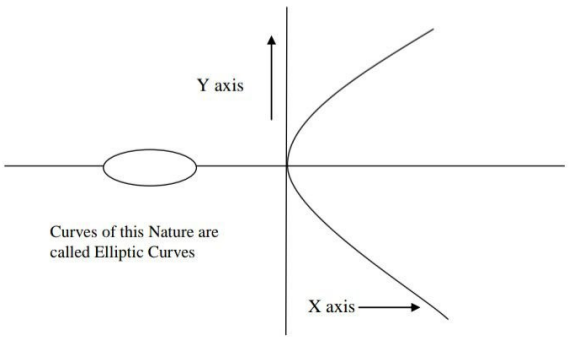
Chapter 3

Elliptic curve cryptography

Elliptic curves cryptography (ECC) is a modern approach to public-key cryptography,the use of elliptic curves in public key cryptography was independently proposed by Koblitz and Miller in 1985.ECC is based on a elliptic curve discrete logarithm problem (ECDLP) and has a high level of security with a shorter keys length compared to the RSA, thus reducing the processing overhead. This chapter begins with a description of elliptic curve, Moreover,the main ECC-based protocols are presented, including ECDLP, ECDH, and ECDSA.The references used in this chapter are:[24, 18, 21, 20, 14, 16, 11, 9, 5, 17]

3.1 Elliptic curve

Elliptic curve are not ellipses. The name is given as elliptic curves because these are described by cubic equations, similar to those used for calculating the circumference of an ellipse. The cubic equations for elliptic curves are of the form $y^2 + axy + by = x^3 + cx^2 + dx + e$ (called Weierstrass equation) , where a, b,c,d and e are real numbers that satisfy some simple conditions. An elliptic curve can be defined over any field (e.g., R, Q, C). However, elliptic curves used in cryptography are mainly defined over finite fields.[9, 10, 16, 11, 14, 17, 18, 21, 24, 22, 20, 15]



3.1.1 Elliptic curve over fields

Definition 3.1. *Let F be a field, and a, b be scalars in F such that the cubic $X^3 + aX + b$ has no multiple roots. An elliptic curve E over a field F is the set of solutions $(X, Y) \in F^2$ to the equation:*

$$Y^2 = X^3 + aX + b$$

plus a "point at infinity" denoted by ∞.

When F is the field of real numbers the condition on the cubic can be expressed in terms of a and b. Let r_1, r_2, r_3 be the roots (maybe complex) of$X^3 + aX + b$, taken together with their multiplicities, i.e., such that

$$X^3 + aX + b = (X - r_1)(X - r_2)(X - r_3)$$

Then it is possible to check that

$$d = (r_1 - r_2)^2(r_1 - r_3)^2(r_2 - r_3)^2 = -(4a^3 + 27b^2)$$

This real number is called the discriminant of the cubic, and the cubic has no multiple roots if and only if this discriminant is nonzero, i.e

$$d = -(4a^3 + 27b^2) \neq 0$$

This condition also guarantees the absence of multiple roots over an arbitrary field F

Example 3.1. *The equation: $Y^2 = X^3 + 3X + 4$. defines an elliptic curve over $\mathbb{Z}_7$ since the discriminant*

$$d = -(4a^3 + 27b^2) = 6 \neq 0$$

The point $(5, 2) \in \mathbb{Z}_7^2$, belongs to this curve since $2^2 \equiv 5^3 + 3 \cdot 5 + 4 \bmod 7$ with both sides being equal to 4.

Example 3.2. *When $\mathbb{F} = \mathbb{R}$ is a field of reals, the graph of an elliptic curve can have two different forms depending on whether the cubic on the right-hand side of $Y^2 = X^3 + aX + b$ has one or three real roots*

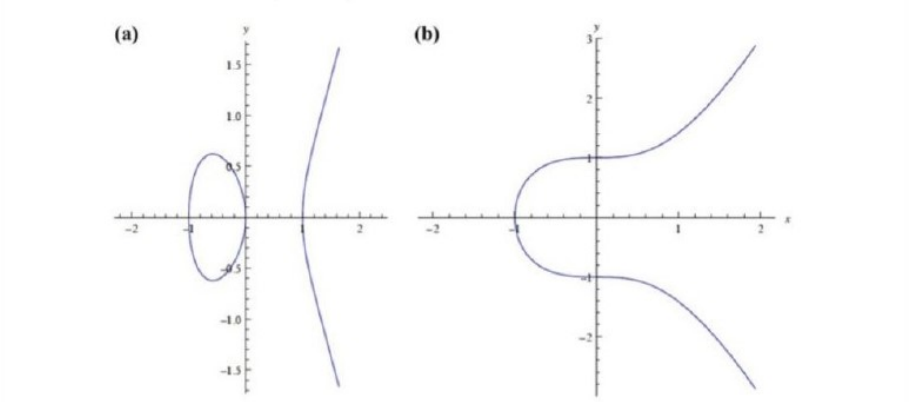


Figure 3.1: Graph elliptic curve $a : y^2 = x(x + 1)(x - 1)$ *and* $b : y^2 = x^3 + 1$

3.1.2 Elliptic curve over finite fields

Let E be an elliptic curve over a finite field $\mathbb{F}_p$ defined by the Weierstrass equation , where $p = q^r$ with p prime number and $r \in N^*$

Definition 3.2. *Let E be an elliptical curve over F_p. From the point of view of cryptography we are interested in elliptic curve group mod p, where p is a prime number. Then E may be described as $4a^3 + 27b^2 \neq 0 \mod p$, where a and b are two positive integers less than the prime number p. Then Ep (a, b) denotes the elliptic group mod p, whose elements (x, y) are pairs of positive integers less then p satisfying $y^2 \equiv x^3 + ax + b \mod p$ together with the point at infinity O.*

Theorem 3.1. ***(Hasse bound)**:Let E be an elliptic curve defined over $\mathbb{F}_p$. Then:*

$$|\#E(\mathbb{F}_p) - (p+1)| \leq 2\sqrt{p}$$

Lemma 3.1. *The group $E(F_p)$ is either cyclic to a product of two cyclic groups,*

$$E(\mathbb{F}_p) \cong \mathbb{Z}/d\mathbb{Z}, where \quad d = \#E(\mathbb{F}_p)$$

$$E(\mathbb{F}_p) \cong \mathbb{Z}/d_1\mathbb{Z} \times \mathbb{Z}/d_2\mathbb{Z}, \quad where \quad d_1|d_2 \quad and \quad d_1|q-1$$

Example 3.3. *Let E be an elliptic curve over a finite field.*

$$E : y^2 = x^3 + 3x + 5$$

over $\mathbb{F}_{17}$, then the set of $\mathbb{F}_{17}$-rational points on E is:

$$\begin{aligned} E(\mathbb{F}_{17}) = \{&(1,14),(1,3),(2,6),(2,11),(4,8),(4,9),\\ &(5,3),(5,14),(6,1),(6,16),(9,8),(9,9),\\ &(10,7),(10,10),(11,14),(11,3),\\ &(12,1),(12,16),(15,12),(15,5),\\ &(16,1),(16,16),\infty\}. \end{aligned}$$

Points of $E(F_{17})$ shown in the figure:

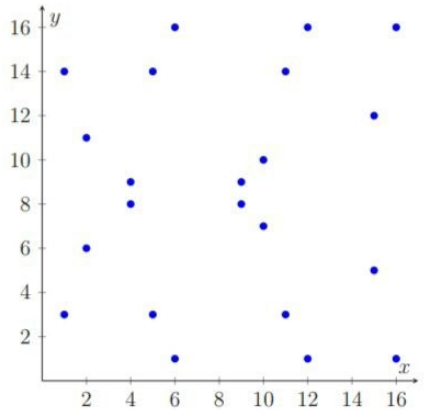


Figure 3.2: An elliptic curve $E : y^2 = x^3 + 3x + 5$

The group $E(\mathbb{F}_{17})$ *is cyclic for all* $P \in E(\mathbb{F}_{17}) \quad (\#E(\mathbb{F}_{17}) = 23)$*, and we have:*

$$|\#E(\mathbb{F}_{17}) - (17+1)| \leq 2\sqrt{17} \quad \textit{and} \quad E(\mathbb{F}_{17}) \cong \mathbb{Z}/23\mathbb{Z}$$

3.1.3 Group Operations on Elliptic Curves

1. **Point multiplication**

   The basic operations of elliptic curve involve point multiplication which is achieved by point addition and point doubling. In the point multiplication a point on the Elliptic Curve say the P is multiplied with a positive integer to obtain the another point of Q on the same Elliptic curve, using the Elliptic curve equations.i.e

   $$Q = KP$$

   $$Let \quad K = 15$$

   $$So; \quad Q = 15P = (2(2(2P + P) + P)) + P$$

   So this example shows that the point multiplication is consummate by using the point addition and the point doubling repeatedly to get the result. This method is named as double and adds method. There are the other efficient methods for the point multiplication such as the NAF (Non – Adjacent Form).

2. **point addition**

   It is the addition of the two points of the elliptic curve, say P and Q , to get another point R on the same Elliptic curve.we have to distinguish two cases: the addition of two distinct points (named point addition) and the addition of one point to itself (named point doubling). **Point Addition** $P + Q$ This is the case where we compute $R = P + Q$ and $P \neq Q$ The construction works as follows: Draw a line through P and Q and obtain a third point of intersection between the elliptic curve and the line. Mirror this third intersection point along the x-axis. This mirrored point is, by definition, the point R. Figure shows the point addition on an elliptic curve over the real numbers.

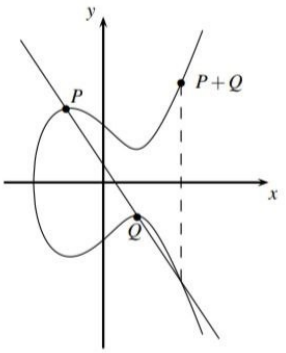


Figure 3.3: Adding points of an ellipic curve

Analytical Explanation: Consider the two distinct points $P(X_P, Y_{P)}$ and $Q(X_Q, Y_Q)$. The slope of line joining these two points is S.

$s = (Y_Q - Y_P)/(X_Q - X_P)$

As we know that the $R = P + Q$, and R is also the point on EC so the coordinates of the $R(X_R, Y_R)$ are calculated by:

$$\begin{cases} X_R = S^2 - X_P - X_Q \\ Y_R = S(X_P - X_R) - Y_P \end{cases}$$

Example 3.4. *Let the elliptic curve be defined over the finite field* $\mathbb{F}_{17}$ *by:*

$$E : y^2 = x^3 + 3x + 5 \pmod{17}$$

Consider the two points on E:

$$P(1, 3), \quad Q(2, 6)$$

We compute the sum $R = P + Q$ *as follows.*

$$s = \frac{y_2 - y_1}{x_2 - x_1} \mod 17$$

$$s = \frac{6 - 3}{2 - 1} = 3$$

$$x_3 = s^2 - x_1 - x_2 \mod 17$$

$$x_3 = 3^2 - 1 - 2 = 6$$

$$y_3 = s(x_1 - x_3) - y_1 \mod 17$$

$$y_3 = 3(1 - 6) - 3 = -18 \equiv 16 \pmod{17}$$

Final result:

$$P + Q = (6, 16)$$

Point Doubling $P + P$ This is the case where we compute P+Q but P = Q. Hence,we can write R = P + P = 2P. We need a slightly different construction here. We draw the tangent line through P and obtain a second point of intersection between this line and the elliptic curve. We mirror the point of the second intersection along the x-axis. This mirrored point is the result R of the doubling. Figure shows the doubling of a point on an elliptic curve over the real numbers.

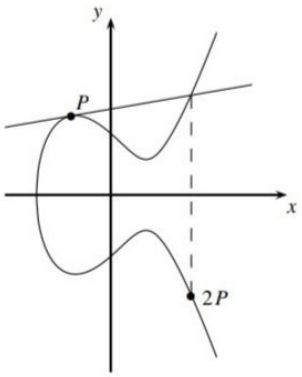


Figure 3.4: Doubling point of an elliptic curve

Analytical Explanation: Consider a point $P(X_P, Y_P)$ where $X_p \neq 0$

Let the $R = 2P$, $R(X_R, Y_R)$. Then the coordinates of $R(X_R, Y_R)$ are calculated by

$$\begin{cases} X_R = S^2 - 2X_P \\ Y_R = S(X_P - X_R) – Y_R \end{cases}$$

S is the curve at the point P and a is the parameter chosen with the Elliptic Curve.

$S = (3X_P^2 + a)/2Y_p$.

Example 3.5. $E : y^2 \equiv x^3 + 2x + 2 \bmod 17$.

We want to compute 2p where P =(5,1).

$2P = P + P = (5, 1) + (5, 1) = (x_3, y_3)$

$$s = \frac{3x^2 + a}{2y}$$

$$= (2 \times 1)^{-1}(3 \times 5^2 + 2)$$

$$= 2^{-1} \times 9$$

$$\equiv 9 \times 9$$

$$\equiv 13 \mod 17$$

$$\begin{cases} x_3 = s^2 - x_1 - x_2 = 132 - 5 - 5 = 159 \equiv 6 \bmod 17 \\ y_3 = s(x_1 - x_3) - y_1 = 13(5 - 6) - 1 = -14 \equiv 3 \bmod 17 \end{cases}$$

2P =(5,1)+(5,1)=(6,3).

For illustrative purposes we check whether the result 2P =(6,3) is actually a point on the curve by inserting the coordinates into the curve equation:

$$y^2 \equiv x^3 + 2x + 2 \bmod 17$$

$$32 \equiv 63 + 2 \times 6 + 2 \bmod 17$$

$$9 = 230 \equiv 9 \bmod 17$$

3. **Inverse of a point** : Even though we made major headway towards the establishment of a finite group, we are not there yet. One thing that is still missing is an identity (or neutral) element O such that:

$$P + O = P$$

for all points P on the elliptic curve. It turns out that there isn't any point (x,y) that fulfills the condition. Instead we define an abstract point at infinity as the neutral element O. This point at infinity can be visualized as a point that is located towards "plus" infinity along the y-axis or towards "minus" infinity along the y-axis. According the group definition, we can now also define the inverse $-P$ of any group element P as:

$$P + (-P) = O$$

The question is how do we find $-P$? If we apply the tangent-and-chord method from above, it turns out that the inverse of the point $P = (x_p, y_p)$ is the point $-P = (x_p, -y_p)$, (the inverse of a point $P(x_p, y_p)$ is now trivial. We simply take the negative of its y coordinate. In the case of elliptic curves over a prime field (the most interesting case in cryptography), this is easily achieved since

$$-y_p \equiv p - y_p \bmod p, hence$$

$$-P = (x_p, p - y_p).$$

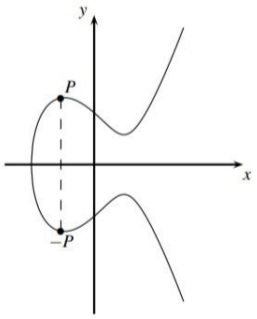


Figure 3.5: Inverse points of an Elliptic Curve

3.2 Protocols based in Elliptic Curve cryptography

In this section, we present the main mathematical foundations and protocols related to Elliptic Curve Cryptography (ECC), which form the core of modern ECC systems. The following topics will be covered:

The Elliptic Curve Discrete Logarithm Problem (ECDLP).

The Key Exchange Protocol (ECDH).

The Digital Signature Algorithm (ECDSA).

here is a general procedure for changing a classical system based on Discrete logarithm into one using elliptic curves:

Classical Finite Field		**Elliptic Curve Setting**
Nonzero numbers mod p	$\Longleftrightarrow$	Points on an elliptic curve
Multiplication mod p	$\Longleftrightarrow$	Elliptic curve addition
1 (multiplicative identity)	$\Longleftrightarrow$	$\mathcal{O}$ (additive identity)
Division mod p	$\Longleftrightarrow$	Subtraction of points
Exponentiation: g^k	$\Longleftrightarrow$	Integer multiplication: $kP = P + \cdots + P$
$p-1$	$\Longleftrightarrow$	$n = \#E(\mathbb{F}_p)$
$a^{p-1} \equiv 1 \pmod{p}$	$\Longleftrightarrow$	$nP = \mathcal{O}$ (Lagrange's theorem)
Discrete logarithm problem:		Elliptic curve discrete logarithm problem
Solve $g^k \equiv h \pmod{p}$ for k	$\Longleftrightarrow$	Solve $kP = Q$ for k

3.2.1 Elliptic Curve Discrete Logarithm Problem

Definition 3.3. *Let E be an elliptic curve over a finite field Fp. We consider P, Q be two points in E(Fp), where P a primitive element of a cyclic subgroup of E (Fp). The elliptic curve discrete logarithm problem (ECDLP) is finding the integer n, where* $1 \leq n \leq \#E$*, such that:*

$$nP = \underbrace{p + p + ... + p}_{n \quad times} = Q$$

By analogy with the discrete logarithm problem for F_p^* *we denote this integer n by* $n = log_P(Q)$*, and we call n the ECDLP of Q with respect to P*

Remark 3.1. *ECDLP is a particular case of DLP, there is no generic algorithm with sub-exponential running time that solves it. Let* $E(F_p)$ *be an elliptic curve over a finite field* F_p*, a point* $P \in E(F_p)$ *of order n, for any point* $Q \in < P >$ *(the subgroup generated by P), the problem is to find out the integer* $k \in [1, n-1]$ *such that* $Q = k \cdot P$*. The integer k is called the discrete log problem of Q to the base P.*

Example 3.6. *Consider the elliptic curve*

$$E : y^2 = x^3 + 2x + 2$$

defined over the finite field $\mathbb{F}_{17}$. *Let*

$$P = (5, 1)$$

be a point on the curve. We compute successive multiples of P*:*

$$P = (5, 1)$$
$$2P = (6, 3)$$
$$3P = (10, 6)$$
$$4P = (3, 1)$$
$$5P = (9, 16)$$
$$6P = (16, 13)$$
$$7P = (0, 6)$$

Suppose that

$$Q = (0, 6)$$

The Elliptic Curve Discrete Logarithm Problem (ECDLP) consists of determining the integer k *such that*

$$Q = kP$$

from the knowledge of P *and* Q. *By observing the successive multiples of* P, *we obtain:*

$$7P = (0, 6) = Q$$

Therefore,

$$k = 7$$

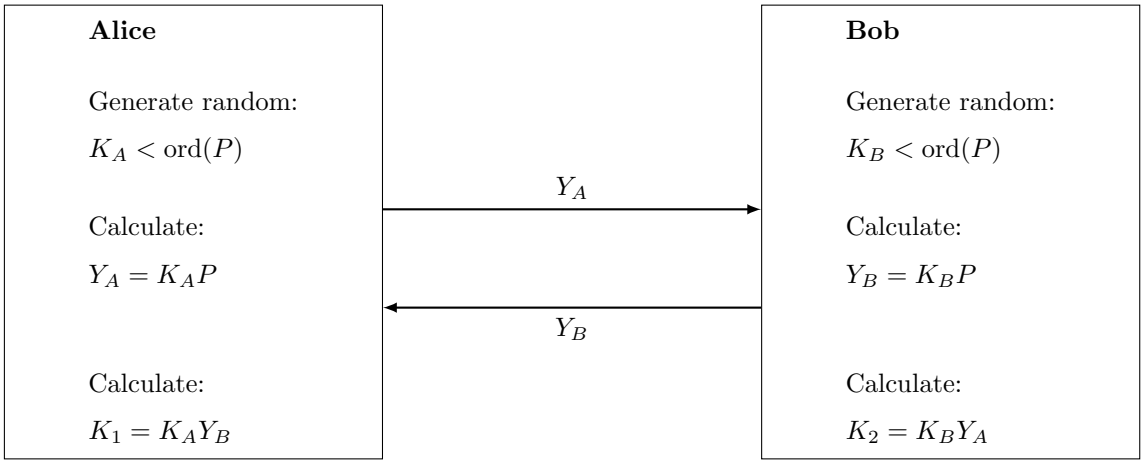
Hence, the discrete logarithm of Q *with respect to* P *is equal to* 7. *This example illustrates that ECDLP can be solved easily for small finite fields by exhaustive search. However, in practical cryptographic systems, the parameters are chosen sufficiently large so that solving ECDLP becomes computationally infeasible.*

3.2.2 Elliptic curve Diffie- Hellman key exchange

In complete analogy to the conventional Diffie-Hellman key exchange (DHKE) presented in section 3.2.1.ECDH is a key agreement protocol that allows two communicating parties to generate a shared secret key.In the way that it is based on the elliptic curve discrete logarithm problem (ECDLP) instead of the discrete logarithm problem (DLP). ECDH Domain Parameters:

1. Alice and Bob Choose a prime p and the elliptic curve $E : y^2 \equiv x^3 + ax + b \mod p$, a generator point $P = (x_P, y_P)$ The prime p.
2. Alice chooses a private random integer $K_A < ord(P)$, and sends $Y_A = K_A P$ to Bob.
3. Bob chooses a private random integer $K_B < ord(P)$, and sends $Y_B = K_B P$ to Alice.
4. Alice computes $K_1 = K_A Y_B$.
5. Bob computes $K_2 = K_B Y_A$.
6. The secret key is $K = K_1 = K_2$.

The ECDHKE is shown in the following figure:



Public parameters:

p : prime number, P : primitive element, $\text{ord}(P)$: order of point P, E : elliptic curve, (Y_A, Y_B) : public keys.

Figure 3.6: Elliptic Curve Diffie-Hellman Key Exchange

Proof. To prove the correctness of the Elliptic Curve Diffie–Hellman (ECDH) protocol, it is sufficient to show that both parties obtain the same shared secret key. Alice computes her public key as:

$$Y_A = k_A P$$

Bob computes his public key as:

$$Y_B = k_B P$$

Then Alice computes:

$$K_1 = k_A Y_B$$

Substituting $Y_B = k_B P$, we obtain:

$$K_1 = k_A(k_B P)$$

Using the associativity of scalar multiplication:

$$K_1 = k_B(k_A P)$$

Since $Y_A = k_A P$, it follows that:

$$K_1 = k_B Y_A$$

But Bob computes:

$$K_2 = k_B Y_A$$

Therefore:

$$K_1 = K_2$$

Hence, both parties derive the same shared secret:

$$K = K_1 = K_2$$

□

Example 3.7. *We consider the public parameters of the ECDH protocol. Let the elliptic curve be:*

$$E : y^2 = x^3 + 2x + 7$$

defined over the finite field $\mathbb{F}_{59}$*. The curve forms a cyclic subgroup:*

$$G = \langle P \rangle$$

generated by the point $P = (3, 10)$ *of order* $\#G = 17$*. The elements of* G *are:*

$$\begin{aligned} P = (3,10), \quad 2P = (7,20), \quad 3P = (31,5), \quad 4P = (38,42), \\ 5P = (20,45), \quad 6P = (9,7), \quad 7P = (1,18), \quad 8P = (28,44), \\ 9P = (28,15), \quad 10P = (1,41), \quad 11P = (9,52), \quad 12P = (20,14), \\ 13P = (38,17), \quad 14P = (31,54), \quad 15P = (7,39), \quad 16P = (3,49), \\ 17P = \infty \end{aligned}$$

ECDH Key Exchange Procedure: *Alice and Bob select secret keys:*

$$k_A = 5 < 17, \qquad k_B = 9 < 17$$

Public key computation:

$$Y_A = k_A P = 5P = (20, 45)$$

$$Y_B = k_B P = 9P = (28, 15)$$

Shared secret computation: *Alice computes:*

$$K = k_A Y_B = 5(28, 15) = 45P = (9, 7)$$

Bob computes:

$$K = k_B Y_A = 9(20, 45) = 45P = (9, 7)$$

Thus, both parties obtain the same shared secret key:

$$K_A = K_B = (9, 7)$$

3.2.3 Elliptic curve ElGamal cryptosystem

Key Generation (Bob): 1. Bob selects an elliptic curve $E(a, b)$ over the finite field $\mathbb{F}_p$.

2. He defines a base point $e_1 = (x_1, y_1) \in E$.
3. He chooses a private key $d \in \mathbb{Z}_n$.
4. He computes the public key:
$$e_2 = d \cdot e_1 = (x_2, y_2)$$
5. Bob publishes (E, p, e_1, e_2) as his public key and keeps d secret.

Encryption (Alice): Alice wants to send a message represented by a point $P \in E$.

1. She selects a random integer $r \in \mathbb{Z}_n$.
2. She computes:
$$C_1 = r \cdot e_1$$
$$C_2 = P + r \cdot e_2$$
3. The cipher-text is (C_1, C_2).

Decryption (Bob): After receiving (C_1, C_2), Bob computes:

$$P = C_2 - d \cdot C_1$$

Correctness:

$$\begin{aligned} P &= C_2 - dC_1 \\ &= P + re_2 - d(re_1) \\ &= P + r(de_1) - dre_1 \\ &= P + \mathcal{O} \\ &= P \end{aligned}$$

where $\mathcal{O}$ is the identity element of the elliptic curve group.

Example 3.8. *We illustrate the ElGamal encryption scheme over an elliptic curve. Let the elliptic curve be:*

$$E : y^2 = x^3 + 2x + 2$$

defined over the finite field $\mathbb{F}_{17}$*. Let the base point be:*

$$e_1 = (5, 1)$$

with order $n = 19$*.* ***Key Generation (Bob):*** *Bob chooses a private key:*

$$d = 7$$

He computes the public key:

$$e_2 = d \cdot e_1 = 7(5, 1) = (0, 6)$$

Thus, Bob's public key is:

$$(E, 17, e_1, e_2)$$

Encryption (Alice): *Alice wants to send a message point:*

$$P = (6, 3)$$

She chooses a random integer:

$$r = 5$$

She computes:

$$C_1 = r \cdot e_1 = 5(5, 1) = (9, 16)$$

$$C_2 = P + r \cdot e_2 = (6, 3) + 5(0, 6)$$

Since:

$$5e_2 = (2, 10)$$

We get:

$$C_2 = (6, 3) + (2, 10) = (7, 6)$$

Thus, cipher-text is:

$$(C_1, C_2) = ((9, 16), (7, 6))$$

Decryption (Bob): *Bob computes:*

$$P = C_2 - d \cdot C_1$$

First:

$$dC_1 = 7(9, 16) = (2, 10)$$

Then:

$$P = (7, 6) - (2, 10)$$

Since subtraction is addition of inverse:

$$P = (7, 6) + (2, 7) = (6, 3)$$

Conclusion: *The recovered plain-text is:*

$$P = (6, 3)$$

which matches the original message.

3.2.4 Elliptic Curve Digital signature algorithm

The shorter bit length of ECC often results in shorter processing time and in shorter signatures. For these reasons, the elliptic curve digital signature algorithm (ECDSA) was standardized in the US by the American National Standards Institute (ANSI) in 1998. The steps in the elliptic curve digital signature algorithm (ECDSA) standard are conceptionally closely related to the DSA scheme (it was discussed in the section 3.2.3). In the ECDSA, Alice generates the signature with her secret-key and Bob verifies the signature with Alice's public-key. This protocol consists of three parts: key generation algorithm, signature generation algorithm and signature verification algorithm. The ECDSA protocol it is shown below which Alice signs the message m and Bob verifies Alice's signature. **key generation algorithm**
Used to generate the public and private key of the users.

1. Use an elliptic curve E over a finite field $\mathbb{F}_p$ with p a prime number, and choose a point $P \in E(\mathbb{F}_p)$ which generates a cyclic group of a prime order n.

2. Choose a random integer d with $0 < d < n$.

3. Compute:
$$Q = dP$$

4. The keys are now:
$$\begin{cases} K_{pub} = (p, n, Q, P) \\ K_{pr} = d \end{cases}$$

Signature generation algorithm

Used by Alice to generate the signature for the message m using private key and hash function h.

1. Select a random or pseudo random integer k, $0 < k < n$.

2. Compute $B = kP = (x_B, y_B)$.

3. Compute $r \equiv x_B \pmod{n}$.

4. If $r = 0$ then go to step 1.

5. Compute hash value of the message $h(m)$.

6. Compute $s \equiv [k^{-1}(h(m) + dr)] \bmod n$.

7. If $s = 0$ then go to step 1.

8. The signature of m is (r, s).

9. Sends the message m and the signature (r, s) to Bob.

Signature verification algorithm

Used by Bob to verify Alice's signature, to accept or reject the message m using the public key and hash function h.

1. Verify that: $0 < r < n$ and $0 < s < n$.

2. Compute the hash value of the message m and convert this bit string $h(m)$ to an integer e.

3. Compute $w \equiv s^{-1} \pmod{n}$.

4. Compute $u_1 \equiv h(m)\, w \pmod{n}$ and $u_2 \equiv rw \pmod{n}$.

5. Compute $A = u_1 P + u_2 Q = (x_A, y_A)$.

6. The verification:

$$\begin{cases} x_A \equiv r \pmod{n} \Rightarrow \text{valid signature.} \\ x_A \not\equiv r \pmod{n} \Rightarrow \text{invalid signature and rejects the signature.} \end{cases}$$

Example 3.9. *Let E be an elliptic curve over the finite field $\mathbb{F}_{29}$ defined by the equation:*

$$E : y^2 \equiv x^3 + 7x + 5 \pmod{29}.$$

Let the point $P = (5, 7) \in E(\mathbb{F}_{29})$ which generates a cyclic subgroup $G \subset E(\mathbb{F}_{29})$ of order:

$$\#G = 11.$$

Alice wants to send a message m to Bob that is to be signed with the ECDSA protocol. The signing and verification process is as follows:

Alice Key generation:

$$\begin{cases} \textit{Choose elliptic curve } E \textit{ with parameters } a = 7,\ b = 5,\ p = 29, \\ \textit{and } P = (5, 7) \textit{ with } n = 11. \\ \textit{Choose private key: } d = 5. \\ \textit{Compute: } Q = dP = 5(5, 7) = (20, 5). \\ \textit{Public key: } (p, a, b, n, P, Q) = (29, 7, 5, 11, (5, 7), (20, 5)). \end{cases}$$

Signature generation:

$$\begin{cases} \textit{Compute hash of message: } h(m) = 20. \\ \textit{Choose ephemeral key: } k = 9,\ 0 < k < n = 11. \\ \textit{Compute: } B = kP = (x_B, y_B) = 9(5,7) = (3,16). \\ r = x_B = 3. \\ s = k^{-1}(h(m) + d \cdot r) \mod 11 = 5(20 + 5.3) \equiv 10 \mod 11. \\ \textit{send to Bob :(m,(r,s)=(m,(3,10)} \end{cases}$$

Bob Verification:

$$\begin{cases} \textit{Verify that: } 0 < r = 3 < 11,\ 0 < s = 10 < 11. \\ w = s^{-1} = 10^{-1} = 10 \mod 11. \\ u_1 = h(m) \cdot w \equiv 20 \cdot 10 \equiv 2 \mod 11. \\ u_2 = r \cdot w \equiv 3 \cdot 10 \equiv 8 \mod 11. \\ A = u_1 P + u_2 Q = 2(5,7) + 8(20,5). \\ verification : x_A \equiv 3 \mod 11 \end{cases}$$

Verification condition:

$$x_A \equiv r \pmod{11} \Rightarrow \textit{valid signature.}$$

3.2.5 ECC encryption/decryption

In this section, we will present how to implement elliptic curve based public key encryption/decryption (asymmetric cryptosystem based on ECC).
Since ECC is an asymmetric cryptography, we require a private-public key pair for to encrypt and decrypt a data. Assume Alice and Bob are the two communicating users, they agree upon a some public parameters are elliptic curve equation E and a primitive point P of cyclic group G. Each user selects a private key and generates public key, let K_A and K_B are private keys and $Y_A = K_A P$ and $Y_B = K_B P$ are public keys of Alice and Bob respectively. **Elliptic Curve Encryption**

1. **Input:** Elliptic curve domain parameters (p, E, P, n), public key Q, Plain text m.
2. **Output:** Cipher text C_m.
3. Represent the plain text m as a point P_m in $E(F_p)$.
4. Select $k \in_R [1, n-1]$.
5. Compute $C_1 = kP$.
6. Compute $C_2 = P_m + kQ$.

7. Return (C_1, C_2).

Elliptic Curve Decryption

Input: Curve domain parameters (p, E, P, n), private key d, ciphertext C_m.

Output: Plaintext m.

1. Compute $P_m = C_2 - dC_1$.
2. Return P_m.

Example 3.10. *Let the elliptic curve be defined over the finite field* $\mathbb{F}_{17}$*:*

$$E : y^2 = x^3 + 2x + 2 \pmod{17}$$

Let $P = (5, 1)$ *be a base point on* E*, generating a cyclic subgroup of order* $n = 19$*. Bob selects a private key:*

$$d_B = 7$$

His public key is computed as:

$$Q_B = d_B P = 7P = (0, 6)$$

Alice encodes her message as a point on the curve:

$$M = (10, 6)$$

She selects a random integer:

$$k = 5$$

The ciphertext components are computed as follows:

$$C_1 = kP = 5P = (9, 16)$$

$$kQ_B = 5Q_B = 5(7P) = 35P = 16P = (10, 11)$$

$$C_2 = M + kQ_B = (10, 6) + (10, 11)$$

$$(10, 6) = -(10, 11)$$

$$C_2 = 0$$

Thus, the cipher-text is:

$$C = ((9, 16), 0)$$

Decryption : *Bob computes:*

$$d_B C_1 = 7(5P) = 35P = 16P = (10, 11)$$

The plain-text is recovered by:

$$M = C_2 - d_B C_1$$

$$M = 0 - (10, 11)$$

$$M = (10, 6)$$

Remark: *The security of this scheme is based on the Elliptic Curve Discrete Logarithm Problem (ECDLP), which makes it computationally infeasible to derive the private key from the public information.*

3.2.6 Comparison between RSA and ECC

Both **RSA** and **Elliptic Curve Cryptography (ECC)** are widely used public-key cryptographic schemes in modern information security systems. Although they serve the same fundamental purpose of enabling secure communication and key exchange, they differ significantly in their mathematical foundations, efficiency, and performance characteristics. The **RSA** cryptosystem is based on the computational difficulty of factoring large composite integers into their prime factors, a problem known as the *Integer Factorization Problem*. In contrast, **ECC** is based on the difficulty of solving the elliptic curve discrete logarithm problem, referred to as the *Elliptic Curve Discrete Logarithm Problem (ECDLP)*, which provides a higher level of security per bit. In terms of key size, **RSA** requires relatively large key lengths to achieve a sufficient security level. Commonly, RSA uses key sizes of 1024 bits or 2048 bits. Conversely, **ECC** achieves equivalent security levels with significantly smaller key sizes, typically ranging from 160 to 256 bits. Regarding computational efficiency, **RSA** operations are generally slower due to the use of large integer arithmetic, which makes it less efficient in constrained environments. On the other hand, **ECC** offers faster computation and reduced processing overhead, making it more suitable for systems with limited computational resources. Furthermore, **ECC** provides advantages in terms of memory usage, energy consumption, and bandwidth efficiency when compared to RSA. As a result, ECC is increasingly preferred in modern applications such as mobile devices, embedded systems, and low-power environments. In conclusion, ECC offers a comparable level of security to RSA while requiring smaller key sizes and providing better overall efficiency, which makes it a more suitable choice for contemporary cryptographic applications. [17, 16]

Symmetric Scheme (bits)	**ECC-based Scheme (bits)**	**RSA / D-H (bits)**
56	112	512
80	160	1024
112	224	2048
128	256	3072
192	384	7680
256	512	15360

Table 3.1: Comparable key sizes for equivalent security

Conclusion

Elliptic Curve Cryptography (ECC) is considered one of the most important modern cryptographic algorithms due to its high efficiency and strong level of security, which makes it essential to study it carefully in order to demonstrate its effectiveness in the world of encryption. Public-key cryptography is an important field of cryptography that enhances security, privacy, and supports the main goals of information security, namely confidentiality, integrity, and availability. This work provided a simple and clear explanation of Elliptic Curve Cryptography, including its definition, mathematical foundations, a brief comparison between ECC and RSA, the advantages of ECC, and some important ECC-based applications such as ECDSA and ECDH.
In conclusion, the aspects related to ECC security, applications, comparison with other methods, and performance can be summarized as follows:

- Key and signature sizes: ECC uses smaller keys compared to RSA, making it more suitable for devices with limited resources.
- Computational complexity: ECC algorithms require less computational power and are therefore faster than RSA algorithms in many operations.
- Security level: ECC is considered to provide a high level of security equal to or greater than RSA while using smaller key sizes.
- Practical performance: In most applications, ECC achieves faster and more efficient performance compared to RSA.
- Suitability for different applications: ECC is an ideal choice for applications requiring small key sizes and fast performance, such as mobile devices and the Internet of Things (IoT), while RSA is widely used in applications requiring high security levels, such as financial transactions.

Bibliography

[1] Balakrishnan, R., & Sridharan, S. (2019). *Discrete Mathematics: Graph Algorithms, Algebraic Structures, Coding Theory, and Cryptography*. CRC Press. https://doi.org/10.1201/9780429486326

[2] Baumslag, G., Fine, B., Kreuzer, M., & Rosenberger, G. (2015). *A Course in Mathematical Cryptography*. De Gruyter. https://doi.org/10.1515/9783110372779

[3] Biggs, N. L. (2008). *Codes: An Introduction to Information Communication and Cryptography*. Springer. https://doi.org/10.1007/978-1-84800-273-9

[4] Gilbert, W. J., & Nicholson, W. K. (2004). *Modern Algebra with Applications* (2nd ed.). Wiley-Interscience.

[5] Hankerson, D. R., Menezes, A. J., & Vanstone, S. A. (2004). *Guide to Elliptic Curve Cryptography*. Springer. https://doi.org/10.1007/b97644

[6] Koblitz, N. *Algebraic Aspects of Cryptography*.

[7] Lidl, R., & Pilz, G. (1998). *Applied Abstract Algebra* (2nd ed.). Springer.

[8] Matsuura, R. *A Friendly Introduction to Abstract Algebra*.

[9] Paar, C., & Pelzl, J. (2010). *Understanding Cryptography: A Textbook for Students and Practitioners*. Springer. https://doi.org/10.1007/978-3-642-04101-3

[10] Rosing, M. (2024). *Elliptic Curve Cryptography for Developers*. O'Reilly Media.

[11] Sharma, P., & Saxena, N. Elliptic curves and their applications in cryptography. *International Journal of Engineering Research*.

[12] Slinko, A. (2015). *Algebra for Applications: Cryptography, Secret Sharing, Error-Correcting, Fingerprinting, Compression*. Springer. https://doi.org/10.1007/978-3-319-21951-6

[13] Smart, N. P. (2016). *Cryptography Made Simple*. Springer. https://doi.org/10.1007/978-3-319-21936-3

[14] Trappe, W., & Washington, L. C. (2020). *Introduction to Cryptography: With Coding Theory* (3rd ed.). Pearson Education.

[15] Winkler, N. *The Discrete Log Problem and Elliptic Curve Cryptography.*

[16] Sahoo, P. K., Chhotray, R. K., Jena, G., & Pattnaik, S. (2013). An implementation of elliptic curve cryptography. *International Journal of Engineering Research & Technology.* Available at: https://www.ijert.org/research/an-implementation-of-elliptic-cryptography-IJERTV2IS1419.pdf

[17] Selikh, B. (2022). *Sur les courbes elliptiques et application à la cryptographie* (Doctoral dissertation, Université de M'sila). Available at: https://univ-msila.dz/

[18] Stinson, D. R., & Paterson, M. B. (2019). *Cryptography: Theory and Practice* (4th ed.). CRC Press.

[19] Judson, T. W. (2016). *Abstract Algebra: Theory and Applications.* LibreTexts. Available at: https://math.libretexts.org/Bookshelves/Abstract_and_Geometric_Algebra/Abstract_Algebra:_Theory_and_Applications_(Judson)

[20] Pote, S., & Katti, J. (2015). Attacks on elliptic curve cryptography discrete logarithm problem (EC-DLP). *International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering*, 3(4), 127–131.

[21] Rabah, K. (2005). *Theory and Implementation of Elliptic Curve Cryptography. Journal of Applied Sciences*, 5(4), 604–633. Available at: https://scialert.net/abstract/?doi=jas.2005.604.633

[22] Diffie, W., & Hellman, M. E. (1976). New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6), 644–654.

[23] Hoffstein, J., Pipher, J., & Silverman, J. H. (2008). *An Introduction to Mathematical Cryptography.* Graduate Texts in Mathematics. Springer-Verlag. Available at: https://doi.org/10.1007/978-0-387-77993-5

[24] Singh, L. D., & Singh, K. M. (2015). Implementation of text encryption using elliptic curve cryptography. *Procedia Computer Science*, 54, 73–82. https://doi.org/10.1016/j.procs.2015.06.009

Arabic Abstract

تتناول هذه المذكرة دراسة التشفير بالمنحنيات الإهليلجية (ECC) باعتباره من أهم أنظمة التشفير بالمفتاح العام الحديثة، لما يوفره من مستوى أمان عالٍ وكفاءة حسابية مقارنة بخوارزمية RSA تم في البداية عرض المفاهيم الجبرية الأساسية المعتمدة في علم التشفير، مثل الزمر والزمر الدورية والحقول المنتهية. كما تم التطرق إلى مسألة اللوغاريتم المنفصل (DLP) ودورها في ضمان أمن أنظمة التشفير. وتركز الدراسة على الجانبين النظري والتطبيقي لـ ECC من خلال تقديم أهم بروتوكولاته، مثل ECDLP و ECDHوECDSA، مع شرح العمليات والخوارزميات المرتبطة بها. وتبرز هذه الدراسة أهمية ECC في تحقيق أمن المعلومات بكفاءة عالية باستعمال مفاتيح صغيرة الحجم.

الكلمات المفتاحية: المنحنيات الإهليلجية، ECC، RSA، التشفير بالمفتاح العام، DLP، ECDLP، ECDH، ECDSA، أمن المعلومات.

English Abstract

This master memory studies Elliptic Curve Cryptography (ECC) as one of the most important modern public-key cryptographic systems, due to its high level of security and computational efficiency compared to the RSA algorithm. The work begins with an introduction to the fundamental algebraic concepts used in cryptography, such as groups, cyclic groups and finite fields. It also discusses the Discrete Logarithm Problem (DLP) and its essential role in securing cryptographic systems. The study focuses on both the theoretical and practical aspects of ECC through the presentation of its main protocols, including ECDLP, ECDH, and ECDSA, together with the associated operations and algorithms. This work highlights the importance of ECC in achieving information security with high efficiency using relatively small key sizes.

Keywords: Elliptic curves, ECC, RSA, public-key cryptography, DLP, ECDLP, ECDH, ECDSA, information security