Republic of Iraq
Ministry of Higher Education and Scientific Research
University of Babylon
Department of Mathematics

Proposed Developments of Elliptic Curves Cryptosystem

A Thesis
Submitted to College of Education
Department of Mathematics
University of Babylon, Babel –Iraq
In Partial Fulfillment of the Requirements for the
Degree of Master of Science in
Mathematics

By

Najlae Falah Hameed

Supervised by

Prof. Dr. Eng. Sattar Bader Sadkhan Almaliky

٢٠٠٥

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

الَّذِينَ آمَنُوا وَتَطْمَئِنُّ قُلُوبُهُمْ بِذِكْرِ اللَّهِ
أَلَا بِــذِكْرِ اللَّــهِ تَطْمَــئِنُّ الْقُلُــوبُ ﴿٢٨﴾
الَّذِينَ آمَنُوا وَعَمِلُوا الصَّالِحَاتِ طُوبَى
لَهُمْ وَحُسْنُ مَآبٍ ﴿٢٩﴾

صَدَقَ اللهُ العَلِيُ العَظِيمُ

سورة الرعد

Supervisor Certification

I certify that this thesis was prepared under my supervision at the Department of Mathematics/College of Education in the University of Babylon as a partial fulfillment of the requirements of the degree of Master of Science in Mathematics.

Signature

Name: Sattar Bader Almaliky
Title: Professor
Date: ١٣ – ١ – ٢٠٠٥

Examination Committee Certificate

We certify that we have read this thesis entitled "***Proposed Developments of Elliptic Curves Cryptosystem***" and as an examining committee, we have examined the student ***Najlae Falah Hameed*** in its contents and that in our opinion, it is adequate with ***Excellence*** standing as a thesis for the degree of Master of Science in Mathematics.

Signature
Name: Dr. Nabil Hashem Kagad
Title: Professor
Date: /   /٢٠٠٥
(*Chairman*)

Signature
Name: Dr. Ali Hussain Bttor
Title: Assistor Professor
Date: /   /٢٠٠٥
(*Member*)

Signature
Name: Dr. Ifticher Mudhar Talb
Title: Teacher
Date: /   /٢٠٠٥
(*Member*)

Signature
Name: Dr. Eng. Sattar Bader Almaliky
Title: Professor
Date: /   /٢٠٠٥
(*Supervisor*)

To my family

for their constant love and support, which made a difficult task possible.

The Researcher

Acknowledgements

I wish to express my gratitude to my thesis supervisor, ***Dr. En. Sattar Bader Sadkhan Almaliky*** (at AL– Nahrain University), for his stimulating guidance and constant encouragement during this thesis work, throughout the months that I worked on the topic.

Also I wish to thank ***My Family***, for all their encouragement and support, without which I would never have completed this work.

Also I would like to express my thanks to the staff of Mathematics Department / College of Education of the Babylon University .

The Researcher

Abstract

One of the main reasons of interest in the Elliptic Curves is that their points form a number of finite abelian group, on which Discrete Logarithm Problem seems to be for more computationally infeasible than the multiplicative group of Finite Fields.

This thesis provides literature review of the main scientific researches in the world and in Iraqi Universities. This review was analyzed in eight different scientific directions.

A Mathematical Background supporting the main aspects of the different Elliptic curve Cryptosystems is well discussed from the mathematical point of view to enhance any interest in these aspects.

A general review of the structure of Elliptic curve Cryptosystem is shown, and some examples of these Cryptosystems were completely explained through the thesis. The different attacking methods were given in this thesis like Exhaustive Search, Parallel Pollard rho Method and Pohlig–Hellman method. These methods give an overview of the attacking efforts required to break the Elliptic curve Cryptosystem.

The main research scientific effort given by this thesis was arranged as follows:

- Proposition to Variant ElGamal Elliptic Curve Cryptosystem with the same computational complexity.
- Proposition to Development of Menezes–Vanstone Elliptic Curve Cryptosystem with more computational complexity
- Proposition of two new algorithms for implementation of ECCss.

All the required mathematical formulas are simulated by using a MATLAB Programming Language Version ٦.٥ to enhance the validity of the developed concepts and ideas. All programs are given in this thesis.

Contents

Chapter ٣ The Elliptic Curve Cryptosystem

Chapter ٥ Discussion and Conclusions

Abbreviations And Mathematical Notations

Abbreviations

DHEK	Diffie – Hellman Exchanging Key
DLP	Discrete Logarithm Problem
DSA	Digital Signature Algorithm
EC	Elliptic Curve
ECDH	Elliptic Curve Diffie – Hellman
ECs	Elliptic Curves
ECCg	Elliptic Curve Cryptography
ECCs	Elliptic Curve Cryptosystem
ECCss	Elliptic Curve Cryptosystems
ECDLP	Elliptic Curve Discrete Logarithm Problem
ECDSA	Elliptic Curve Digital Signature Algorithm
ECMO	Elliptic Curve Massey – Omura
IEEE	Institute of Electrical and Electronics Engineers
IFP	Integer Factorization Problem
MIPS	Million Instructions Per Second
MVECC	Menezes – Vanstone Elliptic Curve Cryptosystem
NIST	National Institute of Standards and Technology
ONB	Optimal Normal Base
PA	Proposed Algorithm
RSA	Rivest– Shamir – Adleman public key encryption scheme
TOF	Trapdoor One – way Function

Mathematical Notations

$<a>$	Cyclic group generated by a
$char(F)$	Characteristic of a field F
F_p	Prime field of order p
$F[x]$	Ring of polynomials over a field F
$deg\ f(x)$	Degree of $f(x)$ where $f(x)$ is a polynomial
$F_{٢}^{m}$	Binary finite field
$\left(\frac{a}{p}\right)$	Legendre symbol
$ord\ (a, n)$	Order of a modulo n
$\Phi(n)$	Euler phi function of n
$[a]_n$	The set of Residue class of a modulo n
$ind_g\ a$	The ***index*** of a to the base g
$\mathbb{Z}_n$	Residue classes modulo n
$A^{٢}$	Affine plane
$P^{٢}(F)$	Projective plane over F
$P^{١}(F)$	Projection line over F
L_∞	Line at infinity.
E	Elliptic curve
O	Point at infinity (on an elliptic curve)
Δ	The discrimiant of E
$j(E)$	The j – invariant of E
P, R, Q	Points on E
$\#E$	The order of curve E
$\mathcal{M}$	Finite set of possible plaintexts
$\mathcal{C}$	Finite set of possible ciphertexts
$\mathcal{K}$	Finite set of possible keys

Appendix A
Proof Of Theorems

Proof of Theorem (٢. ١) (The Chinese Remainder Theorem)

Let $m_{١}, m_{٢}, \ldots, m_r$ be pairwise relatively prime positive integers. Then the system of congruences

$$x \equiv a_{١} \ (mod\ m_{١}),$$
$$x \equiv a_{٢} \ (mod\ m_{٢}),$$
$$\vdots$$
$$x \equiv a_r \ (mod\ m_r),$$

has a unique solution modulo $M = m_{١}\, m_{٢} \ldots m_r$.

Proof //

First, we construct a simultaneous solution to the system of congruences . to do this,

Let $M_k = M/m_k = m_1\, m_2 \ldots m_{k-1}\, m_{k+1} \ldots m_r$.

By hypothesis the m_i are relatively prime in pairs, so that $gcd\ (M_k\ , m_k) = ١$.

Therefore possible to solve the congruence

$M_k\, x_k \equiv ١\ mod\ m_k$; call the unique solution x_k.

Our aim is to prove that

$$y = a_{١}\, M_{١}\, x_{١} + a_{٢} M_{٢}\, x_{٢} + \ldots + a_r\, M_r\, x_r$$

is a simultaneous solution to the given system.

To demonstrate this, it is to be observed that $M_i \equiv ٠\ mod\ m_k$ for $i \neq k$, since $m_k | M_i$ in this case.

Therefore, in the sum for y, all terms except the kth term are congruent to $٠\ mod\ m_k$.

Hence $y \equiv a_k\, M_k\, x_k$

$\equiv a_k\ mod\ m_k$, since $M_k\, x_k \equiv ١\ mod\ m_k$.

This shows that a solution to the given system of congruences exists.

We now show that any two solution are congruent modulo M.

Let $x_{٠}$ and $x_{١}$ both be simulation solution to the system of r congruences. Then, $x_{٠} \equiv x_{١} \equiv a_k\ mod\ m_k$, $k =$ ١, ٢, . . . , r.

So that $m_k|(x_0 - x_1)$

$\Rightarrow M|(x_0 - x_1)$

$\Rightarrow x_0 \equiv x_1 \bmod M.$

This shows that the simulation solution of the system of ***r*** congruences is unique modulo ***M***.

Proof of Theorem (٢. ٢) (Euler's criterion)

Let ***p*** be an odd prime and ***gcd(a, p)*** = ١. then ***a*** is a quadratic residues of ***p*** iff $a^{(p-١)/٢} \equiv ١ \bmod p$

Proof //

⇒) suppose that ***a*** is a quadratic residues of ***p*** ,

so that $x^{٢} \equiv a \bmod p$ has a solution; call it $x_{٠}$.

Since ***gcd (a, p)*** = ١,

then ***gcd*** $(x_{٠}, p)$ = ١,

then $x_0^{p-1} \equiv 1 \bmod p$

then $\left(x_0^2\right)^{(p-1)/2} \equiv 1 \bmod p$

but $a^{(p-1)/2} \equiv \left(x_0^2\right)^{(p-1)/2} \bmod p$

$\equiv x_0^{(p-1)} \bmod p$

$\equiv 1 \bmod p.$

⇐)assume that $a^{(p-١)/٢} \equiv ١ \bmod p$

Since ***gcd (a, p)*** = ١,

then $a^{p-1} \equiv 1 \bmod p$

then $a^{p-1} - 1 \equiv 0 \bmod p$

then $(a^{(p-1)/2} - 1)(a^{(p-1)/2} + 1) \equiv 0 \bmod p$

then either $a^{(p-1)/2} \equiv 1 \bmod p$

or $a^{(p-1)/2} \equiv -1 \bmod p$,but not both . For, if both congruences held simultaneously, then we would have $١ \equiv -١ \bmod p$, or equivalently, $p|2$, which conflicts with our hypothesis.

Since ***a*** is a quadratic nonresidues of ***p*** dose not satisfy $a^{(p-1)/2} \equiv 1 \bmod p$, it must therefore satisfy $a^{(p-1)/2} \equiv -1 \bmod p.$

This observation provides an alternate formulation of ***Euler's criterion*** : the integer ***a*** is a quadratic nonresidues of ***p*** if and only if $a^{(p-1)/2} \equiv -1 \bmod p.$

Proof of Theorem (٢.٣) (Index theorem):

If g is a primitive root modulo n, then $g^x \equiv g^y \pmod n$ iff $x \equiv y \pmod{\Phi(n)}$.

Proof //

$\Leftarrow$) suppose that $x \equiv y \pmod{\Phi(n)}$,

then $x = y + k\Phi(n)$ for some integer k.

therefore $g^x \equiv g^{y+k\Phi(n)} \pmod n$

$\equiv g^y \cdot (g^{\Phi(n)})^k \bmod n$

since g is a primitive root modulo n ,

then $\Phi(n) = ord(g, n)$

then $g^{\Phi(n)} \equiv ١ \bmod n$

then $g^x \equiv g^y \cdot (١)^k \bmod n$

$\equiv g^y \bmod n$.

$\Rightarrow$) suppose that $g^x \equiv g^y \bmod n$

then $g^x \equiv g^y \cdot (١)^k \bmod n$, for some integer k.

since g is a primitive root modulo n ,

then $\Phi(n) = ord(g, n)$

then $g^{\Phi(n)} \equiv ١ \bmod n$

then $g^x \equiv g^y \cdot (g^{\Phi(n)})^k \bmod n$, for some integer k.

$\equiv g^{y+k\Phi(n)} \pmod n$

then $x = y + k\Phi(n)$ for some integer k.

then $x \equiv y \pmod{\Phi(n)}$.

Proof of Theorem (٢.٤)

There are $1+\sum_{x\in F_p}\left(1+\left(\frac{x^3+ax+b}{p}\right)\right)=1+p+\sum_{x\in F_p}\left(\frac{x^3+ax+b}{p}\right)$ points on $E: y^{٢} = x^{٣} + ax + b$, including the point at infinity O, where $\left(\frac{x^3+ax+b}{p}\right)$ is the Legendre symbol.

Proof //

For any $x \in F_p$, x is either the x–coordinate of some point $P \in E(F_p)$ or it is not.

If x is not the x–coordinate of some point, then there does not exist any $y \in F_p$ such that (x, y) satisfies the equation $\left(y^2 = x^3 + ax + b\right)$. That is to say that $x^{٣} + ax + b$ is not a square of some integer and we have

$$1+\left(\frac{x^3+ax+b}{p}\right)=0,\ \text{since}\ \left(\frac{x^3+ax+b}{p}\right)=-1,$$

resulting in no contribution towards $\#(E(F_p))$.

If x is the x–coordinate of some point P, then there exists $y \in F_p$ such that (x, y) satisfies the equation $\left(y^2 = x^3+ax+b\right)$.

Thus, $x^{٣} + ax + b$ is a square of y modulo p and p either divides $x^{٣} + ax + b$ or it does not. If p does not divide $x^{٣} + ax + b$, then $y^{٢} \equiv ٠ \ (mod\ p)$ and by taking square roots, we obtain ٢ such values of y, say $y_{١}$ and $y_{٢}$, yielding two points $(x, y_{١})$ and $(x, y_{٢})$ on $E(F_p)$. Indeed, we have

$$1+\left(\frac{x^3+ax+b}{p}\right)=2,\ \text{since}\ \left(\frac{x^3+ax+b}{p}\right)=1.$$

If p divides $x^{٣} + ax + b$, then $y^{٢} \equiv ٠\ (mod\ p)$ and $y = ٠$, resulting in only one point (x, y) on $E(F_p)$, which is consistent with

$$1+\left(\frac{x^3+ax+b}{p}\right)=1,\ \text{since}\ \left(\frac{x^3+ax+b}{p}\right)=0.$$

Finally, we add ١ to $\#(E(F_p)$ to include the point O and complete the proof.

Proof of Theorem (٢.٥)

Let $E(F_p)$ is an elliptic curve E defined over the finite field F_p have prime order $\#E$, then for all $P \in E(F_p\ )$ and $P \neq ٠$, then P have the order $m = \#\, E$. Then P generates subgroup equal to $E(F_p)$.

Proof //

Let $n = \#\, E$,

Then $m|n$, where m is the order of P [١]

But, by hypothesis, n is prime number,

Then m dose not divide n, iff $m = n$,

Then $m = n$.

[١] Theorem : the order of any element a in abelian group G divided the order of G.

Appendix B
Examples

Example (١)

Let ***G*** = ***{*** ***١*, – *١}***, and the operation is the normal multiplication (•), now ***({*** ***١*, – *١}, •)*** is *abelian group* from the following table:

•	١	–١
١	١	–١
–١	–١	١

Example (٢)

Let ***H*** = ***{*** ***٠*, *٢*, *٤}***, ***G*** = Ƶ , and the operation is $+_{٦}$, then ***({ ٠, ٢, ٤},*** $+_{٦}$ ***)*** is a ***subgroup*** of Ƶ as the following table:

$+_{٦}$	٠	٢	٤
٠	٠	٢	٤
٢	٢	٤	٠
٤	٤	٠	٢

Examples (٣)

١- Show that ***(Ƶ ,+)*** is cyclic group .
٢- Show that ***(***$Ƶ_e$ ***,+)*** is cyclic group [١].

Solution:

١- Since + is normal additive operation ,
then <***a***> = ***{n • a : n*** ∈ ***Ƶ }***
and Ƶ = < ***١***> ***,***
and ***(Ƶ ,+)*** is a cyclic group generated by ***١*** .
٢- Since < ***٢***> = ***{ ٢, ٤ , ٦ , ٨, …},***
then < ***٢***> = $Ƶ_e$ ***,***
and ***(***$Ƶ_e$ ***,+)*** is a cyclic group generated by ***٢*** .

[١] $Ƶ_e$ is the set of even integer numbers ,and Ƶ is the set of all integer numbers.

Example (٤)

The elements of $F_{٢٣}$ are $\{٠,\ ١,\ ٢,\ \dots,\ ٢٢\}$. Let us take the following arithmetic on the field $F_{٢٣}$:

(i) $(١٢+٢٠) \bmod ٢٣=٩$,

(ii) $(٨\cdot ٩) \bmod ٢٣=٣$,

(iii) $(٨^{-١}) \bmod ٢٣=٣$.

Examples (٥)

Construct $F_{٢^٣}$

Solution:

Let $f(x)=x^٣+x+١$, Since $f(١)=f(٠)=١\neq ٠$,

$\therefore$ f is irreducible polynomial .

$\therefore F_{٢^٣}=\{b_٢\omega^٢+b_١\omega+b_٠ : b_٠, b_١, b_٢\in F_٢ \ \&\ \omega^٣+\omega+١=٠\}$.

$b_٢$	$b_١$	$b_٠$
٠	٠	٠
١	٠	٠
٠	١	٠
٠	٠	١
١	١	٠
١	٠	١
٠	١	١
١	١	١

$\therefore F_{٢^٣}=\{٠,\omega^٢,\omega,١,\omega^٢+\omega,\omega^٢+١,\ \omega+١,\ \omega^٢+\omega+١\}$
where $\omega^٣=١+\omega$

In the following tables the addition and multiplication operations in $F_{٢^٣}$ are describe***:***

$+_٢$	٠	١	ω	$\omega^٢$	$١+\omega$	$١+\omega^٢$	$\omega+\omega^٢$	$١+\omega+\omega^٢$
٠	٠	١	ω	$\omega^٢$	$١+\omega$	$١+\omega^٢$	$\omega+\omega^٢$	$١+\omega+\omega^٢$
١	١	٠	$١+\omega$	$١+\omega^٢$	ω	$\omega^٢$	$١+\omega+\omega^٢$	$\omega+\omega^٢$
ω	ω	$١+\omega$	٠	$\omega+\omega^٢$	١	$١+\omega+\omega^٢$	$\omega^٢$	$١+\omega^٢$
$\omega^٢$	$\omega^٢$	$١+\omega^٢$	$\omega+\omega^٢$	٠	$١+\omega+\omega^٢$	١	ω	$١+\omega$
$١+\omega$	$١+\omega$	ω	١	$١+\omega+\omega^٢$	٠	$\omega+\omega^٢$	$١+\omega^٢$	$\omega^٢$
$١+\omega^٢$	$١+\omega^٢$	$\omega^٢$	$١+\omega+\omega^٢$	١	$\omega+\omega^٢$	٠	$١+\omega^٢$	ω
$\omega+\omega^٢$	$\omega+\omega^٢$	$١+\omega+\omega^٢$	$\omega^٢$	ω	$١+\omega^٢$	$١+\omega$	٠	١
$١+\omega+\omega^٢$	$١+\omega+\omega^٢$	$\omega+\omega^٢$	$١+\omega^٢$	$١+\omega$	$\omega^٢$	ω	١	٠

•٢	٠	١	ω	$\omega^{٢}$	$١+\omega$	$١+\omega^{٢}$	$\omega+\omega^{٢}$	$١+\omega+\omega^{٢}$
٠	٠	٠	٠	٠	٠	٠	٠	٠
١	٠	١	ω	$\omega^{٢}$	$١+\omega$	$١+\omega^{٢}$	$\omega+\omega^{٢}$	$١+\omega+\omega^{٢}$
ω	٠	ω	$\omega^{٢}$	$١+\omega$	$\omega+\omega^{٢}$	١	$١+\omega+\omega^{٢}$	$١+\omega^{٢}$
$\omega^{٢}$	٠	$\omega^{٢}$	$١+\omega$	$\omega+\omega^{٢}$	$١+\omega+\omega^{٢}$	ω	$١+\omega^{٢}$	١
$١+\omega$	٠	$١+\omega$	$\omega+\omega^{٢}$	$١+\omega+\omega^{٢}$	$١+\omega^{٢}$	$\omega^{٢}$	١	ω
$١+\omega^{٢}$	٠	$١+\omega^{٢}$	١	ω	$\omega^{٢}$	$١+\omega+\omega^{٢}$	$١+\omega$	$\omega+\omega^{٢}$
$\omega+\omega^{٢}$	٠	$\omega+\omega^{٢}$	$١+\omega+\omega^{٢}$	$١+\omega^{٢}$	١	$١+\omega$	ω	$\omega^{٢}$
$١+\omega+\omega^{٢}$	٠	$١+\omega+\omega^{٢}$	$\omega+\omega^{٢}$	١	ω	$\omega+\omega^{٢}$	$\omega^{٢}$	$١+\omega$

In this representation, $F_{٢^{٣}}$ can be generated by the powers of ***g = (١ ١ ٠):***

$g^{٠} = (٠\ ٠\ ١)$ $\quad g^{١} = (١\ ١\ ٠)$ $\quad g^{٢} = (٠\ ١\ ٠)$ $\quad g^{٣} = (١\ ١\ ١)$

$g^{٤} = (١\ ٠\ ٠)$ $\quad g^{٥} = (١\ ٠\ ١)$ $\quad g^{٦} = (٠\ ١\ ١)$ $\quad g^{٧} = g^{٠} = (٠\ ٠\ ١).$

- ***The multiplicative identity*** for the field is $g^{٠} = (٠\ ٠\ ١)$. The multiplicative inverse of $g^{٥} = (١\ ٠\ ١)$ ***is*** $g^{-٥ \bmod ٧} = g^{٢ \bmod ٧} = (٠\ ١\ ٠)$.

To verify this, see that

$(١\ ٠\ ١)\ (٠\ ١\ ٠) = (\omega^{٢} + ١)\ (\omega) \bmod f(\omega)$

$= (\omega^{٣} + \omega) \bmod f(\omega)$

$= ١$

$= (٠\ ٠\ ١)$, This is the multiplicative identity.

Example (٦)

For the finite field $F_{٢^{٤}}$ use an optimal normal basis representation to compute $(٠\ ١\ ٠\ ٠)\ (١\ ١\ ٠\ ١)$ and $(١\ ٠\ ١\ ٠)^{١٠}$

Solution:

The elements of $F_{٢^{٤}}$ are all binary strings of length ٤, and its of type **I** ONB .

Setup for Multiplication

- $f(x)= x^{٤} + x^{٣} + x^{٢}+ x + ١$. The set of polynomials $\{x,x^{٢}, x^{٤}, x^{٨}\}$ forms a normal basis of $F_{٢^{٤}}$ over $F_{٢}$.

The rows of A are constructed as follows:

Row ٠: $x \bmod f(x) = x = (٠\ ٠\ ١\ ٠)$

Row ١: $x^{٢} \bmod f(x) = x^{٢} = (٠\ ١\ ٠\ ٠)$

Row ٢: $x^{٤} \bmod f(x) = x^{٣} + x^{٢} + x + ١ = (١\ ١\ ١\ ١)$

Row ٣: $x^{٨} \bmod f(x) = x^{٣} = (١\ ٠\ ٠\ ٠)$

Hence the A will be $A=\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$,

and it's inverse will be $A^{-1}=\begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$

The rows of T' are constructed as follows:
Row ٠: $x\,.\,x \; mod \; f(x) = x^{٢} = (٠ \; ١ \; ٠ \; ٠)$
Row ١: $x\,.\,x^{٢} \; mod \; f(x) = x^{٣} = (١ \; ٠ \; ٠ \; ٠)$
Row ٢: $x\,.\,x^{٤} \; mod \; f(x) = x^{٥} \; mod \; f(x) = ١ = (٠ \; ٠ \; ٠ \; ١)$
Row ٣: $x\,.\,x^{٨} \; mod \; f(x) = x^{٩} \; mod \; f(x) = x^{٣} + x^{٢} + x + ١ = (١ \; ١ \; ١ \; ١)$

Hence the T' will be $T'=\begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$,

and then $T = T' \cdot A^{-1} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$

Thus the product terms are:
$l_{٠,٠} = T(٠,٠) = ٠$, $l_{١,٠} = T(٣,٣) = ٠$, $l_{٢,٠} = T(٢,٢) = ١$, $l_{٣,٠} = T(١,١) = ٠$,
$l_{٠,١} = T(١,٠) = ٠$, $l_{١,١} = T(٠,٣) = ٠$, $l_{٢,١} = T(٣,٢) = ١$, $l_{٣,١} = T(٢,١) = ١$,
$l_{٠,٢} = T(٢,٠) = ١$, $l_{١,٢} = T(١,٣) = ١$, $l_{٢,٢} = T(٠,٢) = ٠$, $l_{٣,٢} = T(٣,١) = ٠$,
$l_{٠,٣} = T(٣,٠) = ٠$, $l_{١,٣} = T(٢,٣) = ١$, $l_{٢,٣} = T(١,٢) = ٠$, $l_{٣,٣} = T(٠,١) = ١$.

– ***Multiplication***

The l_{ij} terms which are ١ are: $l_{٠,٢}$, $l_{١,٢}$, $l_{١,٣}$, $l_{٢,٠}$, $l_{٢,١}$, $l_{٣,١}$, and $l_{٣,٣}$. Multiplication is defined by $(a_{٠} \; a_{١} \; a_{٢} \; a_{٣}) \; (b_{٠} \; b_{١} \; b_{٢} \; b_{٣}) = (c_{٠} \; c_{١} \; c_{٢} \; c_{٣})$, where

$c_{٠} = a_{٠}b_{٢} + a_{١}(b_{٢} + b_{٣}) + a_{٢}(b_{٠} + b_{١}) + a_{٣}(b_{١} + b_{٣})$
$c_{١} = a_{١}b_{٣} + a_{٢}(b_{٣} + b_{٠}) + a_{٣}(b_{١} + b_{٢}) + a_{٠}(b_{٢} + b_{٠})$
$c_{٢} = a_{٢}b_{٠} + a_{٣}(b_{٠} + b_{١}) + a_{٠}(b_{٢} + b_{٣}) + a_{١}(b_{٣} + b_{١})$

$c_{٣} = a_{٢}b_{١} + a_{٠}(b_{١} + b_{٢}) + a_{١}(b_{٣} + b_{٠}) + a_{٢}(b_{٠} + b_{٢})$.

Remark[٥٤]

The formula for $c_{١}$ in the multiplication can be obtained by adding a ١ to each subscript in formula for $c_{٠}$ (where the subscripts are reduced modulo ٤). The formula for $c_{٢}$ can be obtained by adding ٢ to each subscript in the formula for $c_{٠}$ and reducing the subscripts modulo ٤. The formula for $c_{٣}$ can likewise be obtained.

Now, $(٠\ ١\ ٠\ ٠)\ (١\ ١\ ٠\ ١) = (c_{٠}\ c_{١}\ c_{٢}\ c_{٣})$, where

$c_{٠} = ٠(٠) + ١(٠ + ١) + ٠(١ + ١) + ٠(١ + ١) = ١$

$c_{١} = ١(١) + ٠(١ + ١) + ٠(١ + ٠) + ٠(٠ + ١) = ١$

$c_{٢} = ٠(١) + ٠(١ + ١) + ٠(٠ + ١) + ١(١ + ١) = ٠$

$c_{٣} = ٠(١) + ٠(١ + ٠) + ١(١ + ١) + ٠(١ + ٠) = ٠$,

then $(٠\ ١\ ٠\ ٠)\ (١\ ١\ ٠\ ١) = (١\ ١\ ٠\ ٠)$.

- ***Exponentiation using Optimal Normal Bases***

The squaring $(a_{٠}\ a_{١}\ a_{٢}\ a_{٣})^{٢} = (a_{٠}\ a_{١}\ a_{٢}\ a_{٣})\ (a_{٠}\ a_{١}\ a_{٢}\ a_{٣}) = (c_{٠}\ c_{١}\ c_{٢}\ c_{٣})$,
Where

$c_{٠} = a_{٠}a_{٢} + a_{١}(a_{٢} + a_{٣}) + a_{٢}(a_{٠} + a_{١}) + a_{٣}(a_{١} + a_{٣}) = a_{٣}^{٢} = a_{٣}$

$c_{١} = a_{١}a_{٣} + a_{٢}(a_{٣} + a_{٠}) + a_{٣}(a_{١} + a_{٢}) + a_{٠}(a_{٢} + a_{٠}) = a_{٠}^{٢} = a_{٠}$

$c_{٢} = a_{٢}a_{٠} + a_{٣}(a_{٠} + a_{١}) + a_{٠}(a_{٢} + a_{٣}) + a_{١}(a_{٣} + a_{١}) = a_{١}^{٢} = a_{١}$

$c_{٣} = a_{٣}a_{١} + a_{٠}(a_{١} + a_{٢}) + a_{١}(a_{٣} + a_{٠}) + a_{٢}(a_{٠} + a_{٢}) = a_{٢}^{٢} = a_{٢}$.

Thus $(a_{٠}\ a_{١}\ a_{٢}\ a_{٣})^{٢} = (a_{٣}\ a_{٠}\ a_{١}\ a_{٢})$ can be calculated with a simple rotation of $(a_{٠}\ a_{١}\ a_{٢}\ a_{٣})$.

$$\begin{aligned}(١\ ٠\ ١\ ٠)^{١٠} &= (١\ ٠\ ١\ ٠)^{٢} * (١\ ٠\ ١\ ٠)^{٢} * (١\ ٠\ ١\ ٠)^{٢} * (١\ ٠\ ١\ ٠)^{٢} * (١\ ٠\ ١\ ٠)^{٢}\\ &= (٠\ ١\ ٠\ ١) * (٠\ ١\ ٠\ ١) * (٠\ ١\ ٠\ ١) * (٠\ ١\ ٠\ ١) * (٠\ ١\ ٠\ ١)\\ &= (٠\ ١\ ٠\ ١)^{٢} * (٠\ ١\ ٠\ ١)^{٢} * (٠\ ١\ ٠\ ١)\\ &= (١\ ٠\ ١\ ٠) * (١\ ٠\ ١\ ٠) * (٠\ ١\ ٠\ ١)\\ &= (١\ ٠\ ١\ ٠)^{٢} * (٠\ ١\ ٠\ ١)\\ &= (٠\ ١\ ٠\ ١) * (٠\ ١\ ٠\ ١)\\ &= (١\ ٠\ ١\ ٠).\end{aligned}$$

In this representation, $F_{٢}^{٤}$ can be generated by the powers of $\alpha = (١\ ١\ ٠\ ٠)$:

$\alpha^{٠} = (١\ ١\ ١\ ١)$ $\alpha^{١} = (١\ ١\ ٠\ ٠)$ $\alpha^{٢} = (٠\ ١\ ١\ ٠)$ $\alpha^{٣} = (٠\ ١\ ٠\ ٠)$

$\alpha^{٤} = (٠\ ٠\ ١\ ١)$ $\alpha^{٥} = (١\ ٠\ ١\ ٠)$ $\alpha^{٦} = (٠\ ٠\ ١\ ٠)$ $\alpha^{٧} = (٠\ ١\ ١\ ١)$

$\alpha^{٨} = (١\ ٠\ ٠\ ١)$ $\alpha^{٩} = (١\ ٠\ ٠\ ٠)$ $\alpha^{١٠} = (٠\ ١\ ٠\ ١)$ $\alpha^{١١} = (١\ ١\ ١\ ٠)$

$\alpha^{١٢} = (٠\ ٠\ ٠\ ١)$ $\alpha^{١٣} = (١\ ١\ ٠\ ١)$ $\alpha^{١٤} = (١\ ٠\ ١\ ١)$ $\alpha^{١٥} = \alpha^{٠} = (١\ ١\ ١\ ١)$.

Example(٧)

For the finite field $F_{٢}^{٧}$ use an optimal normal basis representation to compute $(٠\ ١\ ٠\ ٠\ ١\ ١\ ٠\ ٠)\ (٠\ ١\ ١\ ١\ ٠\ ١\ ٠\ ٠)$ and $(١\ ١\ ١\ ٠\ ٠\ ١\ ٠\ ٠)^{١٢}$

Solution:

The elements of $F_{٢}^{٧}$ are all binary strings of length ٩, and it is not of type **I** or **II** ONB . so there is no ONB representation .

Example(٨)

For the finite field $F_{٢}^{٩}$ use an optimal normal basis representation to compute $(١\ ٠\ ٠\ ٠\ ٠\ ٠\ ٠\ ١\ ٠)(٠\ ١\ ٠\ ٠\ ٠\ ٠\ ١\ ٠\ ٠)$ and $(١\ ٠\ ١\ ٠\ ١\ ٠\ ١\ ٠\ ١)^{١٦}$

Solution:

The elements of $F_{٢}^{٩}$ are all binary strings of length ٧, and it is of type **II** ONB, because of the following:

- ❖ $٢ * ٩ + ١ = ١٩$ which is prime,
- ❖ $\Phi(١٩) = ١٨$, and $ord_{١٩}(٢) = ٢$, then ٢ is primitive in $F_{١٩}$.

So $f(x) = f_m(x)$, and we can compute $f_m(x)$ as follows:

$f_{٠}(x) = ١,$

$f_{١}(x) = x + ١,$

$f_{٢}(x) = x\,f_{١}(x) + f_{٠}(x) = x^{٢} + x + ١,$

$f_{٣}(x) = x\,f_{٢}(x) + f_{١}(x) = x^{٣} + x^{٢} + x + x + ١,$

$= x^{٣} + x^{٢} + ١,$

$f_{٤}(x) = x\,f_{٣}(x) + f_{٢}(x) = x^{٤} + x^{٣} + x + x^{٢} + x + ١,$

$= x^{٤} + x^{٣} + x^{٢} + ١,$

$f_{٥}(x) = x\,f_{٤}(x) + f_{٣}(x) = x^{٥} + x^{٤} + x^{٣} + x + x^{٣} + x^{٢} + ١,$

$= x^{٥} + x^{٤} + x^{٢} + x + ١,$

$f_{٦}(x) = x\,f_{٥}(x) + f_{٤}(x) = x^{٦} + x^{٥} + x^{٣} + x^{٢} + x + x^{٤} + x^{٣} + x^{٢} + ١,$

$= x^{٦} + x^{٥} + x^{٤} + x + ١,$

$f_{٧}(x) = x\,f_{٦}(x) + f_{٥}(x) = x^{٧} + x^{٦} + x^{٥} + x^{٢} + x + x^{٥} + x^{٤} + x^{٢} + x + ١,$

$= x^{٧} + x^{٦} + x^{٤} + ١,$

$f_{٨}(x) = x\,f_{٧}(x) + f_{٦}(x) = x^{٨} + x^{٧} + x^{٥} + x + x^{٦} + x^{٥} + x^{٤} + x + ١,$

$= x^{٨} + x^{٧} + x^{٦} + x^{٤} + ١,$

$f_{٩}(x) = x\,f_{٨}(x) + f_{٧}(x) = x^{٩} + x^{٨} + x^{٧} + x^{٥} + x + x^{٧} + x^{٦} + x^{٤} + ١,$

$= x^{٩} + x^{٨} + x^{٦} + x^{٥} + x^{٤} + x + ١,$

then $f(x) = x^{٩} + x^{٨} + x^{٦} + x^{٥} + x^{٤} + x + ١$, ***i.e.*** $x^{٩} = x^{٨} + x^{٦} + x^{٥} + x^{٤} + x + ١$

The set of polynomials $\{x, x^{٢}, x^{٤}, x^{٨}, x^{١٦}, x^{٣٢}, x^{٦٤}, x^{١٢٨}, x^{٢٥٦}\}$ forms a normal basis of $F_{٢}^{٩}$ over $F_{٢}$.

The rows of A are constructed as follows:

Row ٠: $x \bmod f(x) = x = (٠\ ٠\ ٠\ ٠\ ٠\ ٠\ ٠\ ١\ ٠)$,

Row ١: $x^{٢} \bmod f(x) = x^{٢} = (٠\ ٠\ ٠\ ٠\ ٠\ ٠\ ١\ ٠\ ٠)$,

Row ٢: $x^{٤} \bmod f(x) = x^{٤} = (٠\ ٠\ ٠\ ٠\ ١\ ٠\ ٠\ ٠\ ٠)$,

Row ٣: $x^{٨} \bmod f(x) = x^{٨} = (١\ ٠\ ٠\ ٠\ ٠\ ٠\ ٠\ ٠\ ٠)$,

Row ٤: $x^{١٦} \bmod f(x) = x^{٧}.x^{٩}$

$$\begin{aligned}
&= x^{١٥} + x^{١٣} + x^{١٢} + x^{١١} + x^{٨} + x^{٧}\\
&= x^{٦}(x^{٩} + x^{٧} + x^{٦} + x^{٥} + x^{٢} + x)\\
&= x^{٦}(x^{٨} + x^{٧} + x^{٤} + x^{٣} + ١)\\
&= x^{٥}(x^{٩} + x^{٨} + x^{٥} + x^{٣} + x)\\
&= x^{٥}(x^{٦} + x^{٤} + x^{٣} + ١)\\
&= x^{٢}(x^{٩} + x^{٧} + x^{٦} + x^{٣})\\
&= x^{٢}(x^{٨} + x^{٧} + x^{٥} + x^{٤} + x^{٢} + x + ١)\\
&= x(x^{٩} + x^{٨}) + x^{٧} + x^{٦} + x^{٥} + x^{٣} + x^{٢}\\
&= x(x^{٦} + x^{٥} + x^{٤} + x + ١) + x^{٧} + x^{٦} + x^{٥} + x^{٣} + x^{٢}\\
&= x^{٧} + x^{٦} + x^{٥} + x^{٢} + x + x^{٧} + x^{٦} + x^{٥} + x^{٣} + x^{٢}\\
&= x^{٣} + x\\
&= (٠\ ٠\ ٠\ ٠\ ٠\ ١\ ٠\ ١\ ٠),
\end{aligned}$$

Row ٥: $x^{٣٢} \bmod f(x) = x^{١٦}.x^{١٦}$

$$\begin{aligned}
&= (x^{٣} + x)^{٢}\\
&= x^{٦} + x^{٢}\\
&= (٠\ ٠\ ١\ ٠\ ٠\ ٠\ ١\ ٠\ ٠),
\end{aligned}$$

Row ٦: $x^{٦٤} \bmod f(x) = x^{٣٢}.x^{٣٢}$

$$\begin{aligned}
&= (x^{٦} + x^{٢})^{٢}\\
&= x^{١٢} + x^{٤}\\
&= x^{٣}.x^{٩} + x^{٤}\\
&= x^{١١} + x^{٩} + x^{٨} + x^{٧} + x^{٤} + x^{٣} + x^{٤}\\
&= x^{١١} + x^{٧} + x^{٦} + x^{٥} + x^{٤} + x^{٣} + x + ١\\
&= x^{٢}(x^{٩} + x^{٥} + x^{٤} + x^{٣} + x^{٢} + x) + x + ١\\
&= x^{٢}(x^{٨} + x^{٦} + x^{٣} + x^{٢} + ١) + x + ١\\
&= x(x^{٩} + x^{٧} + x^{٤} + x^{٣} + x) + x + ١\\
&= x(x^{٨} + x^{٧} + x^{٦} + x^{٥} + x^{٣} + ١) + x + ١\\
&= x^{٩} + x^{٨} + x^{٧} + x^{٦} + x^{٤} + x + x + ١\\
&= x^{٧} + x^{٥} + x\\
&= (٠\ ١\ ٠\ ١\ ٠\ ٠\ ٠\ ١\ ٠),
\end{aligned}$$

Row ٧: $x^{١٢٨} \bmod f(x) = x^{٦٤}.x^{٦٤}$

$$\begin{aligned}
&= (x^{٧} + x^{٥} + x)^{٢}\\
&= x^{١٤} + x^{١٢} + x^{٨} + x^{١٢} + x^{١٠} + x^{٦} + x^{٨} + x^{٦} + x^{٢}\\
&= x^{١٤} + x^{١٠} + x^{٢}\\
&= x^{٥}(x^{٩} + x^{٥}) + x^{٢}\\
&= x^{٥}(x^{٨} + x^{٦} + x^{٤} + x + ١) + x^{٢}\\
&= x^{٤}(x^{٩} + x^{٧} + x^{٥} + x^{٢} + x) + x^{٢}\\
&= x^{٤}(x^{٨} + x^{٧} + x^{٦} + x^{٤} + x^{٢} + ١) + x^{٢}
\end{aligned}$$

$$= x^{٣}(x^{٩} + x^{٨} + x^{٧} + x^{٥} + x^{٣} + x) + x^{٢}$$
$$= x^{٣}(x^{٧} + x^{٦} + x^{٤} + x^{٣} + ١) + x^{٢}$$
$$= x(x^{٩} + x^{٨} + x^{٦} + x^{٥} + x^{٢}) + x^{٢}$$
$$= x(x^{٤} + x^{٢} + x + ١) + x^{٢}$$
$$= x^{٥} + x^{٣} + x^{٢} + x + x^{٢}$$
$$= x^{٥} + x^{٣} + x$$
$$= (٠\ ٠\ ٠\ ١\ ٠\ ١\ ٠\ ١\ ٠),$$

Row ٨: $x^{٢٥٦}$ ***mod*** $f(x) = x^{١٢٨} . x^{١٢٨}$

$$= (x^{٥} + x^{٣} + x)^{٢}$$
$$= x^{١٠} + x^{٨} + x^{٦} + x^{٨} + x^{٦} + x^{٤} + x^{٦} + x^{٤} + x^{٢}$$
$$= x^{١٠} + x^{٦} + x^{٢}$$
$$= x(x^{٩} + x^{٥} + x)$$
$$= x(x^{٨} + x^{٦} + x^{٤} + ١)$$
$$= x^{٩} + x^{٧} + x^{٥} + x$$
$$= x^{٨} + x^{٧} + x^{٦} + x^{٤} + ١$$
$$= (١\ ١\ ١\ ٠\ ١\ ٠\ ٠\ ٠\ ١).$$

Thus, $$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

So, $$A^{-1} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

The rows of T' are constructed as follows:

Row ٠: $x . x$ ***mod*** $f(x) = x^{٢} = (٠\ ٠\ ٠\ ٠\ ٠\ ٠\ ١\ ٠\ ٠),$

Row ١: $x . x^{٢}$ ***mod*** $f(x) = x^{٣} = (٠\ ٠\ ٠\ ٠\ ٠\ ١\ ٠\ ٠\ ٠),$

Row ٢: $x . x^{٤} \bmod f(x) = x^{٥} \bmod f(x) = (٠\ ٠\ ٠\ ١\ ٠\ ٠\ ٠\ ٠\ ٠),$

Row ٣: $x . x^{٨} \bmod f(x) = x^{٩} \bmod f(x) = x^{٨} + x^{٦} + x^{٥} + x^{٤} + x + ١$

$= (١\ ٠\ ١\ ١\ ١\ ٠\ ٠\ ١\ ١),$

Row ٤: $x . x^{١٦} \bmod f(x) = x(x^{٣} + x)$

$= x^{٤} + x^{٢}$

$= (٠\ ٠\ ٠\ ٠\ ١\ ٠\ ١\ ٠\ ٠),$

Row ٥: $x . x^{٣٢} \bmod f(x) = x(x^{٦} + x^{٢})$

$= x^{٧} + x^{٣}$

$= (٠\ ١\ ٠\ ٠\ ٠\ ١\ ٠\ ٠\ ٠),$

Row ٦: $x . x^{٦٤} \bmod f(x) = x(x^{٧} + x^{٥} + x)$

$= x^{٨} + x^{٦} + x^{٢}$

$= (١\ ٠\ ١\ ٠\ ٠\ ٠\ ١\ ٠\ ٠)،$

Row ٧: $x . x^{١٢٨} \bmod f(x) = x(x^{٥} + x^{٣} + x)$

$= x^{٦} + x^{٤} + x^{٢}$

$= (٠\ ٠\ ١\ ٠\ ١\ ٠\ ١\ ٠\ ٠),$

Row ٨: $x . x^{٢٥٦} \bmod f(x) = x(x^{٨} + x^{٧} + x^{٦} + x^{٤} + ١)$

$= x^{٩} + x^{٨} + x^{٧} + x^{٥} + x$

$= x^{٧} + x^{٦} + x^{٤} + ١$

$= (٠\ ١\ ١\ ٠\ ١\ ٠\ ٠\ ٠\ ١).$

Thus,
$$T' = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{So,}\quad T = T'.A^{-1} = \begin{bmatrix} 0&1&0&0&0&0&0&0&0\\ 1&0&0&0&1&0&0&0&0\\ 0&0&0&0&1&0&0&1&0\\ 0&0&0&0&0&0&1&0&1\\ 0&1&1&0&0&0&0&0&0\\ 0&0&0&0&0&0&1&1&0\\ 0&0&0&1&0&1&0&0&0\\ 0&0&1&0&0&1&0&0&0\\ 0&0&0&1&0&0&0&0&1 \end{bmatrix}$$

The product terms are:

$l_{٠,٠}$ = *T*(٠,٠) = ٠, $l_{١,٠}$ =*T*(٨,٨) = ١, $l_{٢,٠}$ = *T*(٧,٧) = ٠, $l_{٣,٠}$ = *T*(٦,٦) = ٠,

$l_{٠,١}$ = *T*(١,٠) = ١, $l_{١,١}$ =*T*(٠,٨) = ٠, $l_{٢,١}$ = *T*(٨,٧) = ٠, $l_{٣,١}$ = *T*(٧,٦) = ٠,

$l_{٠,٢}$ = *T*(٢,٠) = ٠, $l_{١,٢}$ =*T*(١,٨) = ٠, $l_{٢,٢}$ = *T*(٠,٧) = ٠, $l_{٣,٢}$ = *T*(٨,٦) = ٠,

$l_{٠,٣}$ = *T*(٣,٠) = ٠, $l_{١,٣}$ =*T*(٢,٨) = ٠, $l_{٢,٣}$ = *T*(١,٧) = ٠, $l_{٣,٣}$ = *T*(٠,٦) = ٠,

$l_{٠,٤}$ = *T*(٠,٠) = ٠, $l_{١,٤}$ =*T*(٣,٨) = ١, $l_{٢,٤}$ = *T*(٢,٧) = ١, $l_{٣,٤}$ = *T*(١,٦) = ٠,

$l_{٠,٥}$ = *T*(١,٠) = ٠, $l_{١,٥}$ =*T*(٤,٨) = ٠, $l_{٢,٥}$ = *T*(٣,٧) = ٠, $l_{٣,٥}$ = *T*(٢,٦) = ٠,

$l_{٠,٦}$ = *T*(٢,٠) = ٠, $l_{١,٦}$ =*T*(٥,٨) = ٠, $l_{٢,٦}$ = *T*(٤,٧) = ٠, $l_{٣,٦}$ = *T*(٣,٦) = ١,

$l_{٠,٧}$ = *T*(٣,٠) = ٠, $l_{١,٧}$ =*T*(٦,٨) = ٠, $l_{٢,٧}$ = *T*(٥,٧) = ١, $l_{٣,٧}$ = *T*(٤,٦) = ٠,

$l_{٠,٨}$ = *T*(٠,٠) = ٠. $l_{١,٨}$ =*T*(٧,٨) = ٠. $l_{٢,٨}$ = *T*(٦,٧) = ٠. $l_{٣,٨}$ = *T*(٥,٦) = ١.

$l_{٤,٠}$ = *T*(٥,٥) = ٠, $l_{٥,٠}$ =*T*(٤,٤) = ٠, $l_{٦,٠}$ = *T*(٣,٣) = ٠, $l_{٧,٠}$ = *T*(٢,٢) = ٠,

$l_{٤,١}$ = *T*(٦,٥) = ١, $l_{٥,١}$ =*T*(٥,٤) = ٠, $l_{٦,١}$ = *T*(٤,٣) = ٠, $l_{٧,١}$ = *T*(٣,٢) = ٠,

$l_{٤,٢}$ = *T*(٧,٥) = ١, $l_{٥,٢}$ =*T*(٦,٤) = ٠, $l_{٦,٢}$ = *T*(٥,٣) = ٠, $l_{٧,٢}$ = *T*(٤,٢) = ١,

$l_{٤,٣}$ = *T*(٨,٥) = ٠, $l_{٥,٣}$ =*T*(٧,٤) = ٠, $l_{٦,٣}$ = *T*(٦,٣) = ١, $l_{٧,٣}$ = *T*(٥,٢) = ٠,

$l_{٤,٤}$ = *T*(٠,٥) = ٠, $l_{٥,٤}$ =*T*(٨,٤) = ٠, $l_{٦,٤}$ = *T*(٧,٣) = ٠, $l_{٧,٤}$ = *T*(٦,٢) = ٠,

$l_{٤,٥} = T(١,٥) = ٠$, $l_{٥,٥} = T(٠,٤) = ٠$, $l_{٦,٥} = T(٨,٣) = ١$, $l_{٧,٥} = T(٧,٢) = ١$,
$l_{٤,٦} = T(٢,٥) = ٠$, $l_{٥,٦} = T(١,٤) = ١$, $l_{٦,٦} = T(٠,٣) = ٠$, $l_{٧,٦} = T(٨,٢) = ٠$,
$l_{٤,٧} = T(٣,٥) = ٠$, $l_{٥,٧} = T(٢,٤) = ١$, $l_{٦,٧} = T(١,٣) = ٠$, $l_{٧,٧} = T(٠,٢) = ٠$,
$l_{٤,٨} = T(٤,٥) = ٠$. $l_{٥,٨} = T(٣,٤) = ٠$. $l_{٦,٨} = T(٢,٣) = ٠$. $l_{٧,٨} = T(١,٢) = ٠$.

$l_{٨,٠} = T(١,١) = ٠$,
$l_{٨,١} = T(٢,١) = ٠$,
$l_{٨,٢} = T(٣,١) = ٠$,
$l_{٨,٣} = T(٤,١) = ١$,
$l_{٨,٤} = T(٥,١) = ٠$,
$l_{٨,٥} = T(٦,١) = ٠$,
$l_{٨,٦} = T(٧,١) = ٠$,
$l_{٨,٧} = T(٨,١) = ٠$,
$l_{٨,٨} = T(٠,١) = ١$.

– ***Multiplication***

The l_{ij} terms which are ١ are: $l_{٠,١}$, $l_{١,٠}$, $l_{١,٤}$, $l_{٢,٤}$, $l_{٢,٧}$, $l_{٣,٦}$, $l_{٣,٨}$, $l_{٤,١}$, $l_{٤,٢}$, $l_{٥,٦}$, $l_{٥,٧}$, $l_{٦,٣}$, $l_{٦,٥}$, $l_{٧,٢}$, $l_{٧,٥}$, $l_{٨,٣}$ and $l_{٨,٨}$.

Multiplication is defined by

$(a_٠\, a_١\, a_٢\, a_٣\, a_٤\, a_٥\, a_٦\, a_٧\, a_٨)\ (b_٠\, b_١\, b_٢\, b_٣\, b_٤\, b_٥\, b_٦\, b_٧\, b_٨) = (c_٠\, c_١\, c_٢\, c_٣\, c_٤\, c_٥\, c_٦\, c_٧\, c_٨)$, where

$c_٠ = a_٠b_١ + a_١(b_٠ + b_٤) + a_٢(b_٤ + b_٧) + a_٣(b_٦ + b_٨) + a_٤(b_١ + b_٢) + a_٥(b_٦ + b_٧) + a_٦(b_٣ + b_٥) + a_٧(b_٢ + b_٥) + a_٨(b_٣ + b_٨)$,

$c_١ = a_١b_٢ + a_٢(b_١ + b_٥) + a_٣(b_٥ + b_٨) + a_٤(b_٧ + b_٠) + a_٥(b_٢ + b_٣) + a_٦(b_٧ + b_٨) + a_٧(b_٤ + b_٦) + a_٨(b_٣ + b_٦) + a_٠(b_٤ + b_٠)$,

$c_٢ = a_٢b_٣ + a_٣(b_٢ + b_٦) + a_٤(b_٦ + b_٠) + a_٥(b_٨ + b_١) + a_٦(b_٣ + b_٤) + a_٧(b_٨ + b_٠) + a_٨(b_٥ + b_٧) + a_٠(b_٤ + b_٧) + a_١(b_٥ + b_١)$,

$c_٣ = a_٣b_٤ + a_٤(b_٣ + b_٧) + a_٥(b_٧ + b_١) + a_٦(b_٠ + b_٢) + a_٧(b_٤ + b_٥) + a_٨(b_٠ + b_١) + a_٠(b_٦ + b_٨) + a_١(b_٥ + b_٨) + a_٢(b_٦ + b_٢)$,

$c_٤ = a_٤b_٥ + a_٥(b_٤ + b_٨) + a_٦(b_٨ + b_٢) + a_٧(b_١ + b_٣) + a_٨(b_٥ + b_٦) + a_٠(b_١ + b_٢) + a_١(b_٧ + b_٠) + a_٢(b_٦ + b_٠) + a_٣(b_٧ + b_٣)$,

$c_٥ = a_٥b_٦ + a_٦(b_٥ + b_٠) + a_٧(b_٠ + b_٣) + a_٨(b_٢ + b_٤) + a_٠(b_٦ + b_٧) + a_١(b_٢ + b_٣) + a_٢(b_٨ + b_١) + a_٣(b_٧ + b_١) + a_٤(b_٨ + b_٤)$,

$c_٦ = a_٦b_٧ + a_٧(b_٦ + b_١) + a_٨(b_١ + b_٤) + a_٠(b_٣ + b_٥) + a_١(b_٧ + b_٨) + a_٢(b_٣ + b_٤) + a_٣(b_٠ + b_٢) + a_٤(b_٨ + b_٢) + a_٥(b_٠ + b_٥)$,

$c_٧ = a_٧b_٨ + a_٨(b_٧ + b_٢) + a_٠(b_٢ + b_٥) + a_١(b_٤ + b_٦) + a_٢(b_٨ + b_٠) + a_٣(b_٤ + b_٥) + a_٤(b_١ + b_٣) + a_٥(b_٠ + b_٣) + a_٦(b_١ + b_٦)$,

$c_٨ = a_٨b_٠ + a_٠(b_٨ + b_٣) + a_١(b_٣ + b_٦) + a_٢(b_٥ + b_٧) + a_٣(b_٠ + b_١) + a_٤(b_٥ + b_٦) + a_٥(b_٢ + b_٤) + a_٦(b_١ + b_٤) + a_٧(b_٢ + b_٧)$.

Thus $(١\ ٠\ ٠\ ٠\ ٠\ ٠\ ٠\ ١\ ٠)(٠\ ١\ ٠\ ٠\ ٠\ ٠\ ١\ ٠\ ٠) = (c_{٠}\ c_{١}\ c_{٢}\ c_{٣}\ c_{٤}\ c_{٥}\ c_{٦}\ c_{٧}\ c_{٨})$, where

$c_{٠}$

$= ١(١)+٠(٠+٠)+٠(٠+٠)+٠(١+٠)+٠(١+٠)+٠(١+٠)+٠(٠+٠)+١(٠+٠)+٠(٠+٠)$

$= ١$,

$c_{١} = ٠+٠+٠+٠+٠+٠+١(٠+١)+٠+١(٠+٠)$

$= ١$,

$c_{٢} = ٠+٠+٠+٠+٠+١(٠+٠)+٠+١(٠+٠)+٠$

$= ٠$,

$c_{٣} = ٠+٠+٠+٠+١(٠+٠)+٠+١(١+٠)+٠+٠$

$= ١$,

$c_{٤} = ٠+٠+٠+١(١+٠)+٠+١(١+٠)+٠+٠+٠$

$= ٠$,

$c_{٥} = ٠+٠+١(٠+٠)+٠+١(١+٠)+٠+٠+٠+٠$

$= ١$,

$c_{٦} = ٠+١(١+١)+٠+١(٠+٠)+٠+٠+٠+٠+٠$

$= ٠$,

$c_{٧} = ٠+٠+١(٠+٠)+٠+٠+٠+٠+٠+٠$

$= ٠$,

$c_{٨} = ٠+١(٠+٠)+٠+٠+٠+٠+٠+٠+٠$

$= ٠$.

Then, $(١\ ٠\ ٠\ ٠\ ٠\ ٠\ ٠\ ١\ ٠)(٠\ ١\ ٠\ ٠\ ٠\ ٠\ ١\ ٠\ ٠) = (١\ ١\ ٠\ ١\ ٠\ ١\ ٠\ ٠\ ٠)$.

- ***Exponentiation using Optimal Normal Bases***

The squaring

$(a_{٠}\ a_{١}\ a_{٢}\ a_{٣}\ a_{٤}\ a_{٥}\ a_{٦}\ a_{٧}\ a_{٨})^{٢} = (a_{٠}\ a_{١}\ a_{٢}\ a_{٣}\ a_{٤}\ a_{٥}\ a_{٦}\ a_{٧}\ a_{٨})(a_{٠}\ a_{١}\ a_{٢}\ a_{٣}\ a_{٤}\ a_{٥}\ a_{٦}\ a_{٧}\ a_{٨})$

$= (c_{٠}\ c_{١}\ c_{٢}\ c_{٣}\ c_{٤}\ c_{٥}\ c_{٦}\ c_{٧}\ c_{٨})$,

Where

$c_{٠} = a_{٠}a_{١} + a_{١}(a_{٠} + a_{٤}) + a_{٢}(a_{٤} + a_{٧}) + a_{٣}(a_{٦} + a_{٨}) + a_{٤}(a_{١} + a_{٢}) + a_{٥}(a_{٦} + a_{٧}) + a_{٦}(a_{٣} + a_{٥}) + a_{٧}(a_{٢} + a_{٥}) + a_{٨}(a_{٣} + a_{٨})$

$= a_{٨}^{٢} = a_{٨}$

$c_{١} = a_{١}a_{٢} + a_{٢}(a_{١} + a_{٥}) + a_{٣}(a_{٥} + a_{٨}) + a_{٤}(a_{٧} + a_{٠}) + a_{٥}(a_{٢} + a_{٣}) + a_{٦}(a_{٧} + a_{٨}) + a_{٧}(a_{٤} + a_{٦}) + a_{٨}(a_{٣} + a_{٦}) + a_{٠}(a_{٤} + a_{٠})$

$= a_{٠}^{٢} = a_{٠}$

$c_{٢} = a_{٢}a_{٣} + a_{٣}(a_{٢} + a_{٦}) + a_{٤}(a_{٦} + a_{٠}) + a_{٥}(a_{٨} + a_{١}) + a_{٦}(a_{٣} + a_{٤}) + a_{٧}(a_{٨} + a_{٠}) + a_{٨}(a_{٥} + a_{٧}) + a_{٠}(a_{٤} + a_{٧}) + a_{١}(a_{٥} + a_{١})$

$= a_{١}^{٢} = a_{١}$

$c_{٣} = a_{٣}a_{٤} + a_{٤}(a_{٣} + a_{٧}) + a_{٥}(a_{٧} + a_{١}) + a_{٦}(a_{٠} + a_{٢}) + a_{٧}(a_{٤} + a_{٥}) + a_{٨}(a_{٠} + a_{١}) + a_{٠}(a_{٦} + a_{٨}) + a_{١}(a_{٥} + a_{٨}) + a_{٢}(a_{٦} + a_{٢})$

$= a_{٢}^{٢} = a_{٢}$

$c_{٤} = a_{٤}a_{٥} + a_{٥}(a_{٤} + a_{٨}) + a_{٦}(a_{٨} + a_{٢}) + a_{٧}(a_{١} + a_{٣}) + a_{٨}(a_{٥} + a_{٦}) + a_{٠}(a_{١} + a_{٢}) + a_{١}(a_{٧} + a_{٠}) + a_{٢}(a_{٦} + a_{٠}) + a_{٣}(a_{٧} + a_{٣})$

$= a_{٣}^{٢} = a_{٣}$

$c_{٥} = a_{٥}a_{٦} + a_{٦}(a_{٥} + a_{٠}) + a_{٧}(a_{٠} + a_{٣}) + a_{٨}(a_{٢} + a_{٤}) + a_{٠}(a_{٦} + a_{٧}) + a_{١}(a_{٢} + a_{٣}) + a_{٢}(a_{٨} + a_{١}) + a_{٣}(a_{٧} + a_{١}) + a_{٤}(a_{٨} + a_{٤})$

$= a_{٤}^{٢} = a_{٤}$

$c_{٦} = a_{٦}a_{٧} + a_{٧}(a_{٦} + a_{١}) + a_{٨}(a_{١} + a_{٤}) + a_{٠}(a_{٣} + a_{٥}) + a_{١}(a_{٧} + a_{٨}) + a_{٢}(a_{٣} + a_{٤}) + a_{٣}(a_{٠} + a_{٢}) + a_{٤}(a_{٨} + a_{٢}) + a_{٥}(a_{٠} + a_{٥})$

$= a_{٥}^{٢} = a_{٥}$

$c_{٧} = a_{٧}a_{٨} + a_{٨}(a_{٧} + b_{٢}) + a_{٠}(a_{٢} + a_{٥}) + a_{١}(a_{٤} + a_{٦}) + a_{٢}(a_{٨} + a_{٠}) + a_{٣}(a_{٤} + a_{٥}) + a_{٤}(a_{١} + a_{٣}) + a_{٥}(a_{٠} + a_{٣}) + a_{٦}(a_{١} + a_{٦})$

$= a_{٦}^{٢} = a_{٦}$

$c_{٨} = a_{٨}a_{٠} + a_{٠}(a_{٨} + a_{٣}) + a_{١}(a_{٣} + a_{٦}) + a_{٢}(a_{٥} + a_{٧}) + a_{٣}(a_{٠} + a_{١}) + a_{٤}(a_{٥} + a_{٦}) + a_{٥}(a_{٢} + a_{٤}) + a_{٦}(a_{١} + a_{٤}) + a_{٧}(a_{٢} + a_{٧})$

$= a_{٧}^{٢} = a_{٧}$

Thus $(a_{٠}\ a_{١}\ a_{٢}\ a_{٣}\ a_{٤}\ a_{٥}\ a_{٦}\ a_{٧}\ a_{٨})^{٢} = (a_{٨}\ a_{٠}\ a_{١}\ a_{٢}\ a_{٣}\ a_{٤}a_{٥}\ a_{٦}\ a_{٧})$ can be calculated with a simple rotation of $(a_{٠}\ a_{١}\ a_{٢}\ a_{٣})$.

$(١\ ٠\ ١\ ٠\ ١\ ٠\ ١\ ٠\ ١)^{١٦} = (١\ ٠\ ١\ ٠\ ١\ ٠\ ١\ ٠\ ١)^{٢} * (١\ ٠\ ١\ ٠\ ١\ ٠\ ١\ ٠\ ١)^{٢} * (١\ ٠\ ١\ ٠\ ١\ ٠\ ١\ ٠\ ١)^{٢} * (١\ ٠\ ١\ ٠\ ١\ ٠\ ١\ ٠\ ١)^{٢} * (١\ ٠\ ١\ ٠\ ١\ ٠\ ١\ ٠\ ١)^{٢} * (١\ ٠\ ١\ ٠\ ١\ ٠\ ١\ ٠\ ١)^{٢} * (١\ ٠\ ١\ ٠\ ١\ ٠\ ١\ ٠\ ١)^{٢} * (١\ ٠\ ١\ ٠\ ١\ ٠\ ١\ ٠\ ١)^{٢}$

$= (١\ ١\ ٠\ ١\ ٠\ ١\ ٠\ ١\ ٠) * (١\ ١\ ٠\ ١\ ٠\ ١\ ٠\ ١\ ٠) * (١\ ١\ ٠\ ١\ ٠\ ١\ ٠\ ١\ ٠) * (١\ ١\ ٠\ ١\ ٠\ ١\ ٠\ ١\ ٠) * (١\ ١\ ٠\ ١\ ٠\ ١\ ٠\ ١\ ٠) * (١\ ١\ ٠\ ١\ ٠\ ١\ ٠\ ١\ ٠) * (١\ ١\ ٠\ ١\ ٠\ ١\ ٠\ ١\ ٠) * (١\ ١\ ٠\ ١\ ٠\ ١\ ٠\ ١\ ٠)$

$= (١\ ١\ ٠\ ١\ ٠\ ١\ ٠\ ١\ ٠)^{٢} * (١\ ١\ ٠\ ١\ ٠\ ١\ ٠\ ١\ ٠)^{٢} * (١\ ١\ ٠\ ١\ ٠\ ١\ ٠\ ١\ ٠)^{٢} * (١\ ١\ ٠\ ١\ ٠\ ١\ ٠\ ١\ ٠)^{٢}$

$= (٠\ ١\ ١\ ٠\ ١\ ٠\ ١\ ٠\ ١) * (٠\ ١\ ١\ ٠\ ١\ ٠\ ١\ ٠\ ١) * (٠\ ١\ ١\ ٠\ ١\ ٠\ ١\ ٠\ ١) * (٠\ ١\ ١\ ٠\ ١\ ٠\ ١\ ٠\ ١)$

$= (٠\ ١\ ١\ ٠\ ١\ ٠\ ١\ ٠\ ١)^{٢} * (٠\ ١\ ١\ ٠\ ١\ ٠\ ١\ ٠\ ١)^{٢}$

$= (١\ ٠\ ١\ ١\ ٠\ ١\ ٠\ ١\ ٠) * (١\ ٠\ ١\ ١\ ٠\ ١\ ٠\ ١\ ٠)$

$= (١\ ٠\ ١\ ١\ ٠\ ١\ ٠\ ١\ ٠)^{٢}$

$= (٠\ ١\ ٠\ ١\ ١\ ٠\ ١\ ٠\ ١).$

Example (٩)

The following are some examples of congruences or incongruences:

$٣٥ \equiv ١١ \pmod{١٢}$ since $١٢ \backslash (٣٥ - ١١)$,

$٣٥ \not\equiv ١٢ \pmod{١١}$ since $١١ / \backslash\ (٣٥\ -\ .$

Example(١٠)

Find the ***quadratic residues*** and ***quadratic nonresidue*** for modulo *٥, ٧, ١١, ١٣,* respectively.

Solution:

- Modulo ٥, the integers ١,٤ are ***quadratic residues*** , while ٢,٣ are ***quadratic nonresidues*** ,since

  $١^٢ \equiv ٤^٢ \equiv ١, \qquad ٢^٢ \equiv ٣^٢ \equiv ٤.$

- Modulo ٧, the integers ١,٢,٤ ***are quadratic residues*** , while ٣,٥,٦ are ***quadratic nonresidues*** ,since

  $١^٢ \equiv ٦^٢ \equiv ١, \qquad ٢^٢ \equiv ٥^٢ \equiv ٤,$
  $٣^٢ \equiv ٤^٢ \equiv ٢.$

- Modulo ١١, the integers ١,٣,٤,٥,٩ are ***quadratic residues*** , while ٢,٦,٧,٨,١٠ are ***quadratic nonresidues*** ,since

  $١^٢ \equiv ١٠^٢ \equiv ١, \qquad ٢^٢ \equiv ٩^٢ \equiv ٤,$
  $٣^٢ \equiv ٨^٢ \equiv ٩, \qquad ٤^٢ \equiv ٧^٢ \equiv ٥,$
  $٥^٢ \equiv ٦^٢ \equiv ٣.$

- Modulo ١٣, the integers ١,٣,٤,٩,١٠,١٢ are ***quadratic residues*** ,

  While ٢,٥,٦,٧,٨,١١ are ***quadratic nonresidues*** .

Now to check the above examples by easy way we look at following tables:

b	$b^٢$	b	$b^٢$	b	$b^٢$	b	$b^٢$
٠	٠	٠	٠	٠	٠	٠	٠
١	١	١	١	١	١	١	١
٢	٤	٢	٤	٢	٤	٢	٤
٣	٤	٣	٢	٣	٩	٣	٩
٤	١	٤	٢	٤	٥	٤	٣
–	–	٥	٤	٥	٤	٥	١٢
–	–	٦	١	٦	٣	٦	١٠
–	–	–	–	٧	٥	٧	١٠
–	–	–	–	٨	٩	٨	١٢
–	–	–	–	٩	٤	٩	٣
–	–	–	–	١٠	١	١٠	٩
–	–	–	–	–	–	١١	٤
–	–	–	–	–	–	١٢	١
Modulo ٥		**Modulo ٧**		**Modulo ١١**		**Modulo ١٣**	

Example (١١)

Let p= ٧ Then,

$$\left(\frac{1}{7}\right)=\left(\frac{2}{7}\right)=\left(\frac{4}{7}\right)=1,$$

$$\left(\frac{3}{7}\right)=\left(\frac{5}{7}\right)=\left(\frac{6}{7}\right)=-1.$$

Example (١٢)

We will compute the values of a^{i}*(mod ١١)* for *i* = ١, ٢, ٣, …, ١٠ in the following table:

a	$a^{٢}$	$a^{٣}$	$a^{٤}$	$a^{٥}$	$a^{٦}$	$a^{٧}$	$a^{٨}$	$a^{٩}$	$a^{١٠}$
١	١	١	١	١	١	١	١	١	١
٢	٤	٨	٥	١٠	٩	٧	٣	٦	١
٣	٩	٥	٤	١	٣	٩	٥	٤	١
٤	٥	٩	٣	١	٤	٥	٩	٣	١
٥	٣	٤	٩	١	٥	٣	٤	٩	١
٦	٣	٧	٩	١٠	٥	٨	٤	٢	١
٧	٥	٢	٣	١٠	٤	٦	٩	٨	١
٨	٩	٦	٤	١٠	٣	٢	٥	٧	١
٩	٤	٣	٥	١	٩	٤	٣	٥	١
١٠	١	١٠	١	١٠	١	١٠	١	١٠	١

$ord_{١١}(١) = ١,$

$ord_{١١}(٢) = ord_{١١}(٦) = ord_{١١}(٧) = ord_{١١}(٨)= ١٠,$

$ord_{١١}(١٠) = ٢.$

Example (١٣)

Let us compute the values of *Φ(n)* for *i* = ١, ٢, ٣,…, ١٠, ١٠٠, ١٠١, ١٠٢, ١٠٣ in the following table:

n	١	٢	٣	٤	٥	٦	٧	٨	٩	١٠	١٠٠	١٠١	١٠٢	١٠٣
Φ(n)	١	١	٢	٢	٤	٢	٦	٤	٦	٤	٤٠	١٠٠	٣٢	١٠٢

Example (١٤)

Determine whether ٧ is ***primitive root*** of ٤٥.

Solution:

Since *gcd (٧, ٤٥) = ١,*

and $٧^{١} \equiv ٧$ *(mod ٤٥),* $٧^{٢} \equiv ٤$ *(mod ٤٥),*

$٧^{٣} \equiv ٢٨$ *(mod ٤٥),* $٧^{٤} \equiv ١٦$ *(mod ٤٥),*

$$٧^{٥} \equiv ٢٢\ (\text{mod } ٤٥), \qquad ٧^{٦} \equiv ١٩\ (\text{mod } ٤٥),$$
$$٧^{٧} \equiv ٤٣\ (\text{mod } ٤٥), \qquad ٧^{٨} \equiv ٣١\ (\text{mod } ٤٥),$$
$$٧^{٩} \equiv ٣٧\ (\text{mod } ٤٥), \qquad ٧^{١٠} \equiv ٣٤\ (\text{mod } ٤٥),$$
$$٧^{١١} \equiv ١٣\ (\text{mod } ٤٥), \qquad ٧^{١٢} \equiv ١\ (\text{mod } ٤٥).$$

Thus, $ord_{٤٥}(٧) = ١٢$.

But $\Phi(٤٥) = ٢٤$ since:

$\underline{\underline{1}},\underline{\underline{2}},3,\underline{\underline{4}},5,6,\underline{\underline{7}},\underline{\underline{8}},9,10,\underline{\underline{11}},12,\underline{\underline{13}},\underline{\underline{14}},15,\underline{\underline{16}},\underline{\underline{17}},18,\underline{\underline{19}},20,21,\underline{\underline{22}},\underline{\underline{23}},24,25,$

$\underline{\underline{26}},27,\underline{\underline{28}},\underline{\underline{29}},30,\underline{\underline{31}},\underline{\underline{32}},33,\underline{\underline{34}},35,36,\underline{\underline{37}},\underline{\underline{38}},39,40,\underline{\underline{41}},42,\underline{\underline{43}},\underline{\underline{44}}.$

Thus ٧ is not a primitive root of ٤٥.

Example (١٥)

Let $n = ٥$. Then there are ***five residue classes modulo*** ٥, namely the set:

$[٠]_{٥} = \{\ldots, -١٥, -١٠, -٥, ٠, ٥, ١٠, ١٥, ٢٠ \ldots\},$
$[١]_{٥} = \{\ldots, -١٤, -٩, -٤, ١, ٦, ١١, ١٦, ٢١ \ldots\},$
$[٢]_{٥} = \{\ldots, -١٣, -٨, -٣, ٢, ٧, ١٢, ١٧, ٢٢ \ldots\},$
$[٣]_{٥} = \{\ldots, -١٢, -٧, -٢, ٣, ٨, ١٣, ١٨, ٢٣ \ldots\},$
$[٤]_{٥} = \{\ldots, -١١, -٦, -١, ٤, ٩, ١٤, ١٩, ٢٤ \ldots\}.$

Example (١٦)

Compute the ***index*** of ١٥ ***base*** ٦ ***modulo*** ١٠٩.

[i.e. $٦^{ind_{٦}١٥}\ (\text{mod } ١٠٩) = ١٥$].

Solution:

To find the ***index***, we just successively perform the computations $٦^{k}\ (\text{mod } ١٠٩)$ for $k = ١, ٢, ٣ \ldots$ until we find a suitable k such that $٦^{k}\ (\text{mod } ١٠٩) = ١٥$:

$$٦^{١} \equiv ٦\ (\text{mod } ١٠٩), \qquad ٦^{٢} \equiv ٣٦\ (\text{mod } ١٠٩),$$
$$٦^{٣} \equiv ١٠٧\ (\text{mod } ١٠٩), \qquad ٦^{٤} \equiv ٩٧\ (\text{mod } ١٠٩),$$
$$٦^{٥} \equiv ٣٧\ (\text{mod } ١٠٩), \qquad ٦^{٦} \equiv ٤\ (\text{mod } ١٠٩),$$
$$٦^{٧} \equiv ٢٤\ (\text{mod } ١٠٩), \qquad ٦^{٨} \equiv ٣٥\ (\text{mod } ١٠٩),$$
$$٦^{٩} \equiv ١٠١\ (\text{mod } ١٠٩), \qquad ٦^{١٠} \equiv ٦١\ (\text{mod } ١٠٩),$$
$$٦^{١١} \equiv ٣٩\ (\text{mod } ١٠٩), \qquad ٦^{١٢} \equiv ١٦\ (\text{mod } ١٠٩),$$
$$٦^{١٣} \equiv ٩٦\ (\text{mod } ١٠٩), \qquad ٦^{١٤} \equiv ٣١\ (\text{mod } ١٠٩),$$
$$٦^{١٥} \equiv ٧٧\ (\text{mod } ١٠٩), \qquad ٦^{١٦} \equiv ٢٦\ (\text{mod } ١٠٩),$$
$$٦^{١٧} \equiv ٤٧\ (\text{mod } ١٠٩), \qquad ٦^{١٨} \equiv ٦٤\ (\text{mod } ١٠٩),$$
$$٦^{١٩} \equiv ٥٧\ (\text{mod } ١٠٩), \qquad ٦^{٢٠} \equiv ١٥\ (\text{mod } ١٠٩).$$

Since $k = ٢٠$ is the smallest positive integer $\ni$ $٦^{٢٠} \equiv ١٥ (\text{mod } ١٠٩)$

Thus $ind_{٦}\, ١٥\ (\text{mod } ١٠٩) = ٢٠$.

Example (١٧)

Compute the ***index*** of ١٦ ***base*** ٧ ***modulo*** ٤٥.

Solution:

To find the ***index***, we just successively perform the computations ٧^{k} ***(mod ٤٥)*** for ***k*** = ١ , ٢ , ٣ ... until we find a suitable ***k*** such that ٦^{k} ***(mod ١٠٩)*** = ١٦:

$$٧^{١} \equiv ٧ \,(mod\ ٤٥), \qquad ٧^{٢} \equiv ٤ \,(mod\ ٤٥),$$
$$٧^{٣} \equiv ٢٨ \,(mod\ ٤٥), \qquad ٧^{٤} \equiv ١٦ \,(mod\ ٤٥),$$

Since ***k*** = ٤ is the smallest positive integer ∋ $٧^{٤} \equiv ١٦ \,(mod\ ٤٥)$,
Thus $ind_{٧}\, ١٦ \,(mod\ ٤٥) = ٤$.

Example (١٨)

An elliptic curve over $F_{٢^{٤}}$ an optimal normal basis representation is used for the elements of $F_{٢^{٤}}$. Consider the non-supersingular curve over $F_{٢^{٤}}$ defined by the equation $y^2 + xy = x^3 + \alpha^3$.

The solution over $F_{٢^{٤}}$ to the elliptic curve equation is:

$$(0,\alpha^9),(\alpha,0),(\alpha,\alpha),(\alpha^3,\alpha^5),(\alpha^3,\alpha^{11}),(\alpha^4,\alpha^3),(\alpha^4,\alpha^7),(\alpha^5,\alpha^3)$$
$$,(\alpha^5,\alpha^{11}),(\alpha^6,0),(\alpha^6,\alpha^6),(\alpha^8,\alpha^3),(\alpha^8,\alpha^{13}),(\alpha^{11},0),(\alpha^{11},\alpha^{11}),$$
$$(\alpha^{12},\alpha^8),(\alpha^{12},\alpha^9),(\alpha^{13},\alpha^2),(\alpha^{13},\alpha^{14}).$$

Since there are ١٩ solutions to the equation in $F_{٢^{٤}}$ the group $E(F_{٢^{٤}})$ has ١٩ + ١ = ٢٠ elements. This group turns out to be a cyclic group of order ٢٠. if we take $G = (\alpha^{٣}, \alpha^{٥})$ and use the addition formulae, we find that

$$1G = (\alpha^3,\alpha^5) \quad 2G = (\alpha^4,\alpha^3) \quad 3G = (\alpha^{13},\alpha^2)$$
$$4G = (\alpha^3,0) \quad 5G = (\alpha^{12},\alpha^8) \quad 6G = (\alpha^8,\alpha^3)$$
$$7G = (\alpha^3,\alpha^5) \quad 8G = (\alpha^4,\alpha^3) \quad 9G = (\alpha^{13},\alpha^2)$$
$$10G = (\alpha^3,0) \quad 11G = (\alpha^{12},\alpha^8) \quad 12G = (\alpha^8,\alpha^3)$$
$$13G = (\alpha^3,\alpha^5) \quad 14G = (\alpha^4,\alpha^3) \quad 15G = (\alpha^{13},\alpha^2)$$
$$16G = (\alpha^3,0) \quad 17G = (\alpha^{12},\alpha^8) \quad 18G = (\alpha^8,\alpha^3)$$
$$19G = (\alpha^3,\alpha^{11}) \quad 20G = O$$

Appendix C

Formulas[١] For The Group Operation

Let E be an elliptic curve with usual Weierstrass equation $E(x , y) = y^{٢} + a_{١}xy + a_{٣}y - x^{٣} - a_{٢}x^{٢} - a_{٤}x - a_{٦} = ٠$, and let $P_{٠} = (x_{٠} , y_{٠}) \in E$, to calculate $-P_{٠}$ we take the line L through $P_{٠}$ and O and find its third point of intersection with E. the line L is given by:

$$L: \; x - x_{٠} = ٠ .$$

Substituting this into the equation for E,

then $E(x_{٠} , y)$ has roots $y_{٠}$ and y'_0 , where $-P_{٠} = (x_0 , y'_0)$.

Writing out

$$E(x_0 , y) = c (y - y_0)(y - y'_0)$$

but $E(x_0 , y) = y^2 + a_1 x_0 y + a_3 y - x_0^3 - a_2 x_0^2 + a_4 x_0 - a_6 = 0$

$$= y^2 + (a_1 x_0 + a_3)y - x_0^3 - a_2 x_0^2 + a_4 x_0 - a_6 = 0$$

now comparing the coefficients of $y^{٢}$ gives c = ١, and then the coefficients of y gives $y'_0 = -y_{٠} - a_{١} x_{٠} - a_{٣}$. this yields

$$-P_{٠} = (x_0 , - y_0 - a_1 x_0 - a_3) .$$

❖ Then for E defined over R or F_p , as we remembered, E has the following formula:

$E : y^2 = x^3 + a_4 x + a_6$, so for any $P = (x_{١} , y_{٢}) \in E$,

$$- P = (x_1 , - y_1 - \underbrace{a_1}_{\substack{\Downarrow \\ 0}} x_1 - \underbrace{a_3}_{\substack{\Downarrow \\ 0}})$$

then $- P = (x_1 , - y_1)$

❖ Then for E defined over $F_{٢}^{m}$, as we remembered, E has the following formula:

$E : y^2 + xy = x^3 + a_2 x^2 + a_6$, so for any $P = (x_{١} , y_{١}) \in E$,

$$- P = (x_1 , - y_1 - \underbrace{a_1}_{\substack{\Downarrow \\ 1}} x_1 - \underbrace{a_3}_{\substack{\Downarrow \\ 0}})$$

[١] All these formulas come from the reference [٤٦] then $-P=(x_1,-y_1-x_1)$.

since the addition and subtraction are equivalent operations in $F_{٢}^{m}$, then

$$-P=(x_1,-y_1-x_1)=(x_1,y_1+x_1)$$

next we derive a formula for the addition law.

Let $P_{١}$ = $(x_{١}, y_{١})$ and $P_{٢}$ = $(x_{٢}, y_{٢})$ be points of E. If $x_{١}$ = $x_{٢}$ and $y_1+y_2+a_1x_2+a_3=0$, then from the above formula $P_{١}+P_{٢}=$ ٠. Otherwise the line L through $P_{١}$ and $P_{٢}$ has an equation of the form:

$$L: y=\lambda x+v.$$

(formulas for λ and v are given below.) Substituting this into the equation for E, then $E(x, \lambda x+v)$ has roots $x_{١}$, $x_{٢}$, $x_{٣}$, where $P_{٣}=(x_{٣},y_{٣})$ is the third point of $L\cap E$. (where $P_{١}+P_{٢}+P_{٣}=$ ٠) while Writing out

$$E(x,\lambda x+v)=c(x-x_1)(x-x_2)(x-x_3)$$

$$=c(x^2-x_2x-x_1x+x_1x_2)(x-x_3)$$

$$=c(x^3-x_2x^2-x_1x^2+x_1x_2x-x_3x^2+x_2x_3x+x_1x_3x-x_1x_2x_3)$$

$$=c(x^3-(x_2+x_1+x_3)x^2+x_2x_3x+x_1x_3x-x_1x_2x_3)$$

but

$$E(x,\lambda x+v)=(\lambda x+v)^2+a_1x(\lambda x+v)+a_3(\lambda x+v)-x^3-a_2x^2-a_4x-a_6$$

$$=\lambda^2x^2+2\lambda xv+v^2+a_1\lambda x^2+a_1vx+a_3\lambda x+a_3v-x^3-a_2x^2-a_4x-a_6$$

$$=-x^3+(\lambda^2+a_1\lambda-a_2)x^2+(2\lambda v+a_1v+a_3\lambda-a_4)x+v^2+a_3v-a_6$$

so from comparing the coefficients of $x^{٣}$ and $x^{٢}$ yields $c=-١$ and

$$(x_2+x_1+x_3)=(\lambda^2+a_1\lambda-a_2).$$

Then $x_3=\lambda^2+a_1\lambda-a_2-x_2-x_1$, and substituting into the equation for L gives $y_{٣}=\lambda x_{٣}+v$. finally, to find $P_{١}+P_{٢}=-P_{٣}$, we apply the negation formula found above to $P_{٣}$. These are summarized as follows:

<u>Group Law Algorithm</u> Let E be an elliptic curve given by a Weierstrass equation

$$E: y^{٢}+a_{١}xy+a_{٣}y=x^{٣}+a_{٢}x^{٢}+a_{٤}x+a_{٦}.$$

(a) let $P_{٠}=(x_{٠},y_{٠})\in E$. then

$$-P_{٠}=(x_0,-y_0-a_1x_0-a_3).$$

Now let

$$P_{١}+P_{٢}=P_{٣} \text{ with } P_i=(x_i,y_i)\in E.$$

(b) if $x_{١}=x_{٢}$ and $y_1+y_2+a_1x_2+a_3=0$, then

$$P_{١}+P_{٢}=٠.$$

Otherwise, let

$$\lambda = \begin{cases} \dfrac{y_2 - y_1}{x_2 - x_1} & \textit{if} \quad x_2 \neq x_1 \\[2ex] \dfrac{3x_1^2 + 2a_2x_1 + a_4 - a_1y_1}{2y_1 + a_1x_1 + a_3} & \textit{if} \quad x_2 = x_1 \end{cases}$$

and $$v = \begin{cases} \dfrac{y_1x_2 - y_2x_1}{x_2 - x_1} & \textit{if} \quad x_2 \neq x_1 \\[2ex] \dfrac{3x_1^3 - a_4x_1 + 2y_1^2}{2y_1 + a_1x_1 + a_3} & \textit{if} \quad x_2 = x_1 \end{cases}$$

(then $y = \lambda x + v$ is the line through ***P*١** and ***P*٢** , or tangent to ***E*** if ***P*١ = *P*٢**.)

(c) ***P*٣ = *P*١ + *P*٢** is given by

$$x_3 = \lambda^2 + a_1\lambda - a_2 - x_2 - x_1$$
$$y_3 = -(\lambda + a_1)x_3 - v - a_3 .$$

❖ Then for ***E*** defined over ***R*** or F_p , as we remembered, ***E*** has the following formula:

$$E : y^2 = x^3 + a_4x + a_6 ,$$

so for any ***P*١ = (*x*١ , *y*١)** and ***P*٢ = (*x*٢ , *y*٢)** ∈***E*** such that ***x*١ ≠*x*٢** ,
and let ***P*٣ = (*x*٣ , *y*٣) = *P*١ + *P*٢** ,

then $$x_3 = \lambda^2 + \underset{\substack{\Downarrow\\ 0}}{\underbrace{a_1}}\lambda - \underset{\substack{\Downarrow\\ 0}}{\underbrace{a_2}} - x_1 - x_2$$

$$= \lambda^2 - x_1 - x_2 ,$$

$$y_3 = -(\lambda + \underset{\substack{\Downarrow\\ 0}}{\underbrace{a_1}})x_3 - v - \underset{\substack{\Downarrow\\ 0}}{\underbrace{a_3}}$$

$$= \frac{-y_2 + y_1}{x_2 - x_1}x_3 - \frac{y_1x_2 - y_2x_1}{x_2 - x_1}$$

$$= \frac{-y_2x_3 + y_1x_3 - y_1x_2 + y_2x_1}{x_2 - x_1}$$

$$= \frac{-y_1x_2 + y_1x_1 + y_2x_1 - y_2x_3 - y_1x_1 + y_1x_3}{x_2 - x_1}$$

$$= \frac{-y_1(x_2 - x_1) + y_2(x_1 - x_3) - y_1(x_1 - x_3)}{x_2 - x_1}$$

$$= \frac{-y_1(x_2 - x_1) + (y_2 - y_1)(x_1 - x_3)}{x_2 - x_1}$$

$$= \frac{-y_1(x_2 - x_1)}{(x_2 - x_1)} + \frac{(y_2 - y_1)(x_1 - x_3)}{x_2 - x_1}$$

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

when $P = (x_١ , y_١)$, such that $x_١ \neq ٠$, then $٢P = (x_٢ , y_٢)$, where

$$\lambda = \frac{3x_1^2 + 2\overbrace{a_2}^{\substack{0\\ \Uparrow}} x_1 + a_4 - \overbrace{a_1}^{\substack{0\\ \Uparrow}} y_1}{2y_1 + \underbrace{a_1}_{\substack{\Downarrow\\ 0}} x_1 + \underbrace{a_3}_{\substack{\Downarrow\\ 0}}}$$

then $\lambda = \dfrac{3x_1^2 + a_4}{2y_1}$

$$v = \frac{3x_1^3 - a_4 x_1 + 2y_1^2}{2y_1 + \underbrace{a_1}_{\substack{\Downarrow\\ 0}} x_1 + \underbrace{a_3}_{\substack{\Downarrow\\ 0}}}$$

$$= \frac{3x_1^3 - a_4 x_1 + 2y_1^2}{2y_1}$$

$$x_2 = \lambda^2 + \underbrace{a_1}_{\substack{\Downarrow\\ 0}}\lambda - \underbrace{a_2}_{\substack{\Downarrow\\ 0}} - x_1 - x_1$$

$$= \lambda^2 - 2x_1 ،$$

$$y_2 = -(\lambda + \underbrace{a_1}_{\substack{\Downarrow\\ 0}})x_2 - v - \underbrace{a_3}_{\substack{\Downarrow\\ 0}}$$

$$= -(\frac{3x_1^2 + a_4}{2y_1})x_2 - \frac{3x_1^3 - a_4 x_1 + 2y_1^2}{2y_1}$$

$$= \frac{-3x_1^2 x_2 - a_4 x_2 - 3x_1^3 + a_4 x_1 - 2y_1^2}{2y_1}$$

$$= \frac{-3x_1^3 - 3x_1^2 x_2 + a_4 x_1 - a_4 x_2 - 2y_1^2}{2y_1}$$

$$= \frac{-3x_1^2(x_1 - x_2) + a_4(x_1 - x_2) - 2y_1^2}{2y_1}$$

$$=\frac{(-3x_1^2+a_4)(x_1-x_2)-2y_1^2}{2y_1}$$

$$=\frac{(-3x_1^2+a_4)}{2y_1}(x_1-x_2)-\frac{2y_1^2}{2y_1}$$

$$=\lambda(x_1-x_2)-y_1$$

❖ Then for E defined over $F_{٢}^{m}$, as we remembered, E has the following formula:

$$E: y^2+xy=x^3+a_2x^2+a_6,$$

so for any $P_{١}=(x_{١}, y_{١})$ and $P_{٢}=(x_{٢}, y_{٢})$ $\in E$ such that $x_{١}\neq x_{٢}$,
and let $P_{٣}=(x_{٣}, y_{٣})=P_{١}+P_{٢}$,

$$x_3=\lambda^2+\underbrace{a_1}_{\substack{\Downarrow\\ 1}}\lambda-a_2-x_1-x_2,$$

since the addition and subtraction are equivalent operations in $F_{٢}^{m}$
, then $x_3=\lambda^2+\lambda+a_2+x_1+x_2\cdot$

$$y_3=-(\lambda+\underbrace{a_1}_{\substack{\Downarrow\\ 1}})x_3-v-\underbrace{a_3}_{\substack{\Downarrow\\ 0}}$$

since the addition and subtraction are equivalent operations in $F_{٢}^{m}$
, then

$$y_3=\left(\frac{y_2+y_1}{x_2+x_1}+1\right)x_3+\frac{y_1x_2+y_2x_1}{x_2+x_1}$$

$$=\frac{y_2x_3+y_1x_3}{x_2+x_1}+x_3\,\frac{y_1x_2+y_2x_1}{x_2+x_1}$$

$$=\frac{y_2x_3+y_1x_3+x_2x_3+x_1x_3+y_1x_2+y_2x_1}{x_2+x_1}$$

$$=\frac{y_2x_1+y_2x_3+y_1x_1+y_1x_3+x_2x_3+y_1x_2+x_1x_3+y_1x_1}{x_2+x_1}$$

$$=\frac{y_2(x_1+x_3)+y_1(x_1+x_3)+x_2(x_3+y_1)+x_1(x_3+y_1)}{x_2+x_1}$$

$$=\frac{(x_1+x_3)(y_2+y_1)+(x_2+x_1)(x_3+y_1)}{x_2+x_1}$$

$$=\frac{(x_1+x_3)(y_2-y_1)}{(x_2-x_1)}+(x_3+y_1)$$

$$=\lambda(x_1+x_3)+x_3+y_1$$

when $P = (x_{١}, y_{١})$, such that $x_{١} \neq ٠$, then ٢$P = (x_{٢}, y_{٢})$, where

$$\lambda = \frac{\overset{\overset{1}{\Uparrow}}{3}x_1^2 + \overset{\overset{0}{\Uparrow}}{2}x_1 + \overbrace{a_4}^{\overset{0}{\Uparrow}} - \overbrace{a_1}^{\overset{1}{\Uparrow}}y_1}{\underset{\underset{0}{\Downarrow}}{2}y_1 + \underbrace{a_1}_{\underset{1}{\Downarrow}}x_1 + \underbrace{a_3}_{\underset{0}{\Downarrow}}}$$

then $\lambda = \dfrac{x_1^2 + y_1}{x_1}$

$$= x_1 + \frac{y_1}{x_1}$$

$$v = \frac{\overset{\overset{1}{\Uparrow}}{3}x_1^3 - \overbrace{a_4}^{\overset{0}{\Uparrow}}x_1 + \overset{\overset{0}{\Uparrow}}{2}y_1^2}{\underset{\underset{0}{\Downarrow}}{2}y_1 + \underbrace{a_1}_{\underset{1}{\Downarrow}}x_1 + \underbrace{a_3}_{\underset{0}{\Downarrow}}}$$

$$= \frac{x_1^3}{x_1} = x_1^2$$

$$x_2 = \lambda^2 + \underbrace{a_1}_{\underset{1}{\Downarrow}}\lambda + a_2 + x_1 + x_1$$

$$= \lambda^2 + \lambda + a_2,$$

$$y_2 = (\lambda + \underbrace{a_1}_{\underset{1}{\Downarrow}})x_2 + v + \underbrace{a_3}_{\underset{0}{\Downarrow}}$$

$$= (x_1 + \frac{y_1}{x_1} + 1)x_2 + x_1^2$$

$$= \frac{x_1^2\ x_2 + y_1x_2 + x_1x_2 + x_1^3}{x_1}$$

$$= \frac{x_1^3 + x_1^2x_2 + y_1x_2 + x_1x_2}{x_1}$$

$$= \frac{x_1^3}{x_1} + \frac{(x_1^2 + y_1 + x_1)x_2}{x_1}$$

$$= x_1^2 + (\lambda + 1)x_2$$

Chapter One
Introduction

(١.١) Introduction

Classically, the making and breaking of secret codes has usually been confined to diplomatic and military practices. With the growing quantity of digital data stored and communicated by electronic data–processing systems, organizations in both the public and commercial sectors have felt the need to protect information from unwanted intrusion. indeed, the widespread use of electronic funds transfers has made privacy a pressing concern in most financial transactions. There has thus been a recent surge of interest by mathematicians and computer scientists in cryptography [١] which is the science of making the communications unintelligible to all except authorized parties. Cryptography is the one of the known practical means for protecting information transmitted through public communications networks, such as those using telephone lines, microwaves or satellites [١].

The history of cryptography is long and fascinating, and a very significant turning point came in ١٩٧٦ when two researchers from Stanford, Whitfield Diffie and Martin Hellman, published their paper “New Directions in Cryptography” [٢]. In that paper, they introduced the revolutionary concept of public key cryptography [٣]. EC systems as applied to cryptography were first proposed in ١٩٨٥ independently by Neal Koblitz from the University of Washington, and Victor Miller, who was then at IBM, Yorktown Heights [٤], [٥], [٦]. In contrast, of the history of public key cryptosystem, the elliptic curves are not new to the field of Number Theory – they have been

[١] The word cryptography comes from the Greek kryptos meaning hidden and graphein meaning to write.

studied and scrutinized for most of this past century. But the application of elliptic curves to the field of cryptography is a recent phenomenon, beginning barely ١٠ years ago [٧]. The Elliptic Curve Cryptosystem (ECCs) was thus created [٣]. The proposed ECCss are analogs of existing schemes. it is possible to define analogs of public-key cryptosystems that are based on the Discrete Logarithm Problem (DLP) (such as ElGamal encryption and the DSA for instance) [٨]. Since then, numerous researchers and developers have spent years researching the strength of ECCs and improving techniques for its implementation [٣].

Today, it offers those looking for a smaller and faster public key cryptosystem, a practical and secure technology, even for the most constrained environments. based on the Discrete Logarithm Problem in the multiplicative group of a finite field [٣]. The security of this scheme based on intractability of DLP in the multiplicative group of a large finite field [٦], [٩]. Elliptic Curve Discrete Logarithm Problem (ECDLP) appears to be much harder than the DLP in the other group, such as Digital Signature Algorithm (DSA) problem and Integer Factorization Problem (IFP) (i.e Rivest-Shamir-Adleman public key encryption scheme (RSA). Hence ECCs can match the security of the other cryptosystems while using smaller key [٩]. For example, *Table (١. ١)*, illustrates that [١٠].

Table (١. ١): Key Size

Time to break (in ***MIPS***[٢] years)	***RSA*** key size (in bits)	***ECC*** key size (in bits)	***RSA/ECC*** key size ratio
$١٠^{٤}$	٥١٢	١٦٠	٥: ١
$١٠^{٨}$	٧٦٨	١٣٢	٦: ١
$١٠^{١١}$	١٠٢٤	١٦٠	٧: ١
$١٠^{٢٠}$	٢٠٤٨	٢١٠	١٠: ١
$١٠^{٧٨}$	٢١٠٠٠	٦٠٠	٣٥: ١

[٢] Million Instructions Per Second.

(١. ٢) Literature Review

As we said, EC Cryptographic schemes were proposed independently in ١٩٨٥ by Neal Koblitz and Victor Miller. They are the EC analogues of schemes based on the DLP where the underlying group is the group of points on an EC defined over a finite field .

After the above work which is the outline of the creating of ECCs, all researchers[٣] which are working in the subject of ECCs , distribute their works −as we think− into eight directions, let us arrange them as follows:

(١.٢. ١) *"Mathematical Contributions of Elliptic Curve Cryptography"*. This direction is very important to the reader who is beginning in study of ECCg , let us classify this direction into three ways, as follows:

(١.٢.١.١) *"Mathematics of Elliptic Curve"*. where this way concludes the papers which are talking about the mathematics of EC only without cryptography like [١١], where in ٢٠٠١, this paper explains the group structure of elliptic curves, and giving the equations for adding and doubling points on ECs. A basic background of arithmetic of elliptic curves over fields of characteristic ٢ is given in [١٢] in ٢٠٠١. This will include formulas for the group operations on an EC over a field of characteristic ٢. This paper will also discuss the underlying field operations, particularly multiplication. And [١٠], where in ٢٠٠١ also, this paper is an overview of the mathematics of elliptic curves. [١٠] is better than [١٢] and [١١] because it is more illustrative.

(١.٢.١.٢) *"Mathematics of Elliptic Curve Cryptography"* where this way conclude the papers which are talking about the mathematics of Elliptic Curve Cryptography ECCg, and all these papers assumed that the reader has a basic understanding of cryptography and additionally, has a basic understanding of abstract

[٣] when we search in the internet for looking for the researches in our object, we find a lot of these researches. But not all have a date for published it, we will mention the research which have a date for published it algebra and elementary number theory, like [١٣], where in ٢٠٠٢, this paper is as a tutorial to ECCg, and the motivation of this paper is to provide the reader with facts about the use of Elliptic Curves in Cryptography, so lots of (beautiful) mathematics is not included. And in ٢٠٠٢ and ٢٠٠٤, [١٤] and [١٥] respectively present a notes on ECCg, −as we think− [١٣] is more illustrative than [١٤] and [١٥] in details.

(١.٢.١.٣) *"Mathematics of Elliptic Curve and Mathematics of Elliptic Curve Cryptography"* where, in ٢٠٠١ [١٦] is talking in the same time about the mathematics of EC and of ECCg and also talking on attack of ECCg[٤].

(١.٢. ٢) *"Elliptic Curve Cryptography over finite field"* in this direct we find six papers, when the reader read this subject, he may thinking about it as a mathematics of EC or as mathematics of ECCg, but in the fact these papers which we mention it discussion only one field and how implementing the EC on it in the Cryptography. let us classifying this direct into two ways, as follows:

(١.٢.٢.١) *"Elliptic Curve Cryptography over Binary Field"* like [١٧], where in ١٩٩٨, this paper described a method to represent points on ECs over $F_{٢}^{m}$, in the context of ECCss, using n bits. And Since n bits are necessary to represent a point in the general case of a cryptosystem over $F_{٢}^{m}$, this paper is talking about it. And [١٨], where in ٢٠٠٠, this paper discusses an architecture of finite field $F_{٢}^{٢m}$ multiplier using normal basis representation. This paper proposed architecture offers lower computational time and lower complexity architecture compared with other architecture. And [١٩], where in ٢٠٠٠, this paper describes in brief about the implementation of ECCg in Embedded System, and also discusses the different ways of implementation in Galois Field and suggests using Optimal Extension Field, so This paper gives brief way of currently known ways of implementations in embedded systems. And [٢٠], where

[٤] all the above papers are published for help the reader to understanding the main object of mathematics of Elliptic Curve Cryptography.

in ٢٠٠٣, this paper discusses generic arithmetic for arbitrary binary fields in the context of ECCg, and the implementation of this paper takes a general approach to implement binary field arithmetic and does not depend on any choice of field size or field polynomial, also this paper shows that this implementation is faster than other implementations for arbitrary field sizes. And [٢١], where in ٢٠٠٣, this paper demonstrate an efficient implementation of an EC scalar multiplication over $\boldsymbol{F}_{٢}^{m}$, using one of the leading reconfigurable computers available on the market.

(١.٢.٢.٢) *"Elliptic Curve Cryptography over Prime Field"* like [٢٢], where in ٢٠٠٣, this paper presents a study of various algorithms for performing underlying field arithmetic and point representation useful in software implementations of ECCg over prime fields as well as binary fields. So there is not enough papers talking about this direct .

(١.٢. ٣) *"Implementation of Elliptic Curve Cryptosystem "*. In this direction, we find six papers, these papers may be similar to proposition which we have in chapter four, since it designs and suggests in more of these papers, like [٩], where in ١٩٩٨, this paper design a processor to perform ElGamal ECCs scheme for non-supersinguler curve over $F_{٢}^{١٥٥}$ for having efficient method of multiplication, adding, squaring and inversing field elements. And [٢٣], where in ١٩٩٨, this paper concerned with the implementation of ECCs and is a demonstration of that implementation. Additional pertinent examples, illustrations and supporting computer programs are included to present a self-contained work. And [٢٤], where in ٢٠٠٢, this paper presents an efficient implementation of an ECCs and discuss theoretical and practical aspects of the ECCg, and this paper find a system is flexible enough to implement. And [٢٥], where in ٢٠٠٣, this paper describes the system which is called Elliptic Curve Integrated Encryption Scheme, which was proposed by Abdalla, Bellare and Rogaway , where this scheme has the advantage that its security has been proven under standard assumptions only. And also in ٢٠٠٣, [٢٦] provide a short overview on how to use the library to create EC digital signatures, for help the reader, how he great this system. And [٢٧], where in ٢٠٠٤, this paper presents a scalable mobile cryptosystem, which installs a group key and an EC private/public key pair in each device to enable both symmetric key and public key cryptography, the system user should choose the best appropriate protocol, by taking into account the level of security range required and the operational cost that the user is willing to accept.

(١.٢. ٤) *"Perform Comparison of Elliptic Curve Cryptosystem with other well-known Public-Key Cryptography"*. We find in this subject five papers and all these have the result which say that the ECCs was the better than any other publickey widely use, like [٨], where in ١٩٩٧, this paper made, a comparison between an application based on an ECCs and one based on RSA, and it finds that an ECCs over $\boldsymbol{F}_{٢}^{٢١٦٠}$ offers the same security as ١٠٢٤-bit RSA. And [٢٨], where in ١٩٩٧, this paper provides an overview of current public-key cryptographic systems and compares the ECCs with other well known systems, and find that ECCs offers significant efficiency savings due to its added strength-per-bit. And [٢٩], where in ١٩٩٩, this paper specifies public-key cryptographic schemes based on ECCg, and this paper published to help ensure ongoing detailed analysis of ECCs by cryptographers by clearly, completely, and publicly specifying baseline techniques. And [٣٠], where in ٢٠٠٠, this paper clarifies whether ECCg is better suited to be used with smart cards than the nowadays widely used ١٠٢٤-bit RSA, this paper shows that the ECCs is faster than RSA for decrypting and signing messages. And [٣١], where in ٢٠٠٠, this paper describes ECCss, this paper find that although ECCss security has not been completely evaluated, it is expected to come into widespread use in various fields in the future because of its compactness and high performance when it is hardware-implemented, thus ECCss are fast and can be implemented with less hardware.

(١.٢. ٥) *"The Elliptic Curve Discrete Logarithm Problem"*. Where this comes in all papers which we mention it, but the papers which talk about this direction only are organized as follows: [٣٢], where in ١٩٩٣, this paper demonstrates the reduction of the ECLP to the logarithm problem in the multiplicative group of an extension of the underlying finite field, for the class of supersingular Elliptic Curve. And [٧], where in ١٩٩٧, this paper interests in public-key cryptosystems that use the ECDLP to establish security, and to illustrate this idea, it discuss a sample implementation of the EC analogue of the ElGamal cryptosystem, and this paper find that the idea of implementing digital signature/identification schemes in the form of smart cards has quickly gained momentum. And [٣٣], where in ١٩٩٧, and this paper update in ٢٠٠٠, this paper provides an overview of the three hard mathematical problems which provide the basis for the security of publickey cryptosystems used today: the Integer Factorization Problem, the Discrete Logarithm Problem , and the ECDLP, and find that the ECCs as a mature technology and are now implementing it for widespread deployment. And [٣٤], where in ٢٠٠١, this paper presents the Silver-Pohlig-Hellman algorithm for computing Discrete Logarithms in a group. And find that the existence of the Silver-Pohlig-Hellman algorithm means that an encrypter wishing to be certain of the security of an encryption on a "random" EC $\boldsymbol{E}$ must know that $\boldsymbol{\#E(F_p{}^n)}$ is divisible by at least one large prime $\boldsymbol{q}$. And [٣٥], where in ٢٠٠١, this paper describes the state-of-the-art in algorithms for solving the ECDLP, and this paper constructs the table which is summarizes the known attacks on the ECDLP and If an EC is selected that meets all the requirements in this table, then the ECDLP is intractable against all known attacks.

(١.٢. ٦) *"Point on an Elliptic Curve"*. The count of points on an EC is very important, since this counting inters in the searching about suitable EC for cryptography, like [٣٦], where in ١٩٩٢, this paper explain the implementation and improvements of three algorithms which are designed to determine the number of points on an EC $\boldsymbol{E}$ over a finite field, this paper for application only. And [٣٧], where in ٢٠٠١, this paper presents two very straightforward techniques for counting the number of points on an EC over a finite field. The first amounts to the evaluation of a number of Legendre symbols, and the second, Shanks' "baby-step, giant-step" algorithm, is based on a slightly subtler idea. The first method is very practical for fields of very small size, while the baby-step, giant-step algorithm work quickly as long as the field is of moderate size, and in the part two of this paper given a method that work for much larger field , and this paper with two parts is for application only.

(١.٢. ٧) *“Hyperelliptic Curve Cryptosystem”*[٥]. In this direction we find three papers only, and its organized as follows: [٣٨], where in ٢٠٠٠, this paper outlines the issues arising from the implementation of a commercially secure Hyperelliptic Curve Cryptosystem over Optimal Extensions Fields as a followup to the student-devised EC product, and shown that they can be a feasible encryption technique. And [٣٩], where in ٢٠٠١, this paper development of computer architectures for the different algorithms needed for Hyperelliptic Curve Cryptosystem, in particular, development of suitable algorithms to implement the necessary field operations in hardware. And [٤٠], where in ٢٠٠١, this paper introduce Hyperelliptic Curves as a basis for a cryptographic system, specifically, details concerning implementation, performance, and known attacks will be addressed, this paper also conclude that Hyperelliptic Curve Cryptosystems are currently inefficient, but that further research is necessary.

(١.٢. ٨) *“Searching for suitable Elliptic Curves for Cryptography”*. In this direction we find only two papers, and its organized as follows: [٣], where Academic Year ٢٠٠٠/٢٠٠١, this paper uses two methods to search for curves for the ECCs. The first method involves the definition of an EC over a number of fields and then its reduction modulo prime ideals, the second method defines an EC over a small finite field and then considers it over extensions of the small field, results show that both methods are effective in searching for a suitable EC for the Cryptosystem. And [٤١], where in ٢٠٠١, this paper compares methods for choosing EC for Cryptographic application, this includes three methods[٦], the first one is counting the number of points on randomly chosen curve using Satoh’s Algorithm, which is the winner because of its order of magnitude speed superiority, and the second is constructing the curve using Weil Theorem, which is the winner if one needs to work in $\boldsymbol{F}_{p}$ for large prim $\boldsymbol{p}$, and the last method is constructing the curve using the methods

[٥] In ١٩٨٩, Neal Koblitz introduced the idea of using mathematical objects called hyperelliptic curves as a basis for cryptosystems.

[٦] This paper did not go into the details of how the methods work.

of Complex Multiplication, which is third because its order of magnitude slowness compares with other methods.

In Iraqi Universities, the researches of ECCs –in the Mathematical Science or Computer Science– are not enough[٧], with the publish searches in the new world, although they have some important ideas such as [٤٢] (٢٠٠٣) where in thesis proved a new theorem and corollary that can contribute in facilitating the computation process and give methods to conclude the number of the points of the EC without their calculation and using one of these theorems to propose the algorithm for random search about a curve with specified number of points, as well as design a program for implementing process of multiplying a point with a constant number. and some important propositions to enhance of ECCs, such as [٤٣] publishes in (٢٠٠٤) where this thesis proved a new theorem that can contribute in facilitating the computational process and give method to conclude that each point in the EC −except the point at infinity− is generator point, if the order of this Elliptic is prime, also this thesis provides four proposed methods as a modification of ElGamal Cryptosystem with the EC, as well as, design a program for implementing process of these proposed methods. And also in the University of Technology, there are two searches [٤٤],[٤٥], uses the EC –but not ECCs– to primality testing in ٢٠٠٠ and ٢٠٠١ respectively.

(١.٣) The Aim of the Thesis

١. Study some of algebraic preliminaries and some of the basic concepts of the number theory which we use to defined the group of EC and the DLP.
٢. study the details of the ECCss and the attacks of it.
٣. Try to vary the ElGamal ECCs with same complexity.
٤. Try to develop the encryption and decryption of Menezes–Vanstone scheme.
٥. Try to benefit from the Diffie–Hellman Exchanging Key (DHEK).

[٧] They represented only in University of Technology.

(١.٤) The Overview of the Thesis

In the chapter two, we shall elaborate on the mathematical background required for this project.

In Chapter three, we shall give an introduction to the mechanism of the ECCss with an examples, and the three importance attacks of it also with examples.

In chapter four, we Try to vary the ElGamal ECCs with same complexity. Also we shall try to develop the encryption and decryption of Menezes–Vanstone scheme in the three propositions. Finally, we shall try to benefit from the DHEK for use this key as a secret key in two suggestion methods.

In the end of this thesis, we presented three appendixes, the first one contains the proof of the theorems which we mention in chapter two, the second contains the examples of concepts which we mention in chapter two, while the third contains the derivation the formulas for the EC group operation .

Chapter Two

Mathematical Background

(٢. ١) Introduction

Much of today's cryptography is based heavily on the use of modern abstract algebra, where the theory of groups, fields and many of the abstract data types used in cryptography are in fact structures taken from this area of mathematics. In particular ECs defined over prime field and binary field are generally used in cryptography [٢٢]. Ttherefore in this chapter, we shall explain some of algebraic preliminaries such as group and its laws, field, ... ,etc. Also we will present some of the basic concepts of the number theory which we use this in thesis .

Finally, we will express the definition of the equation of an EC over a field and explain all the operations laws of the group of points on ECs over all fields.

(٢. ٢) Basic Algebra

All mathematical concepts necessary for understanding the studies on ECs will be provided in this section.

To understand the structure of groups and fields, let us define firstly some simple, yet important properties of binary relations. Let S be a set of elements of type $s_{١}, s_{٢} ... s_k$ and let ○ be a binary operation defined on S, then the following laws are defined [٣٨]:

– closure: if $s_i, s_j \in S$ then $(s_i \circ s_j) \in S \quad \forall i, j \in\{ ١, ٢,...,k\}$

– commutativity: $s_i \circ s_j = s_j \circ s_i \quad \forall i, j \in\{ ١, ٢,...,k\}$

– associativity: $(s_i \circ s_j) \circ s_r = s_i \circ (s_j \circ s_r) \quad \forall i, j , r \in\{ ١, ٢,...,k\}$

– identity: $\exists e \in S, \forall s_i \in S, \; s_i \circ e = s_i = e \circ s_i \quad \forall i \in\{ ١, ٢,...,k\}$

– inverse: $\forall s_i \in S, \exists \; s_j, \quad s_i \circ s_j = s_i \circ s_j = e \quad \forall i, j \in\{ ١, ٢,...,k\}$

And, if a second binary operation + which also defined on ***S*** is add, then:

– distributivity: $s_i \circ (s_j + s_r) = s_i \circ s_j + s_i \circ s_r \ \forall i, j, r \in$ *{ ١, ٢,...,k}.*

A set of tools are available with which it is possible to define the properties of the next important structure, that of a group. Groups can be further specified as abelian groups or subgroups of larger groups and the following definitions are also included.

Definition (٢. ١) [٤٦]

A ***Group*** ***(G, ○)*** is a set ***G*** together with a binary operator ○, with the following properties : closure, identity, inverse and associativity .

Definition (٢. ٢) [٤٦]

An ***Abelian Group*** ***(G, ○)*** is a Group with the extra property of being commutative.***(*** i.e. $a \circ b = b \circ a \quad \forall \ a, b \in G$***).***

Definition (٢. ٣) [٣٨]

Given a subset $H \subseteq G$, ***(H, ○)*** is a ***subgroup*** of the group ***(G, ○)*** if ***(H, ○)*** is a group itself.

Definition (٢. ٤) [٤٦]

Let ***a*** be an element of a multiplicative group [١] ***G*** . The elements a^r , where ***r*** is an integer, form a subgroup of ***G***, called the ***subgroup generated by a***, denoted by <***a***>. (i.e. <***a***> = $\{a^r , r \in \mathbb{Z}\}$ ***a***∈ ***G***).

Remarks (٢. ١):

- A subgroup ***H*** is a ***cyclic*** if ***H*** is generated by one element [٤٦],[٤٧].
- The group ***(G, ○)*** is cyclic if there exist an element $a \in G$ such that <***a***> = ***G*** , in this case ***G cyclic group generator by a*** [٤٦] ,[٤٧].
- ***Order Of Group*** is the number of elements in the group [٤٦],[٤٧].
- Let ***G*** be any group, and $a \in G$ then the ***order*** of ***a*** is ***n*** iff $a^n = e$.[٢] [٤٧]

[١] If the binary operation of a group is denoted by **.** , then the identity of a group is denoted by ١ or ***e***; this group is said to be a ***multiplicative group*** [٤٦]

[٢] ***e*** is the identity element in ***G*** .

The most relevant and important of the elementary constructions associated with abstract algebra are fields, in particular finite ones. This is because the elements of these fields are used to embed the plaintext upon. Then, a series of well–defined actions are taken on these elements, outputting elements of the same field from which the ciphertext can be extrapolated [٣٨].

Let us begin with a discussion of fields themselves including definitions of the characteristic of such a field, and its algebraic closure.

Definition (٢.٥) [٣٨]

A ***field*** ***(F, · , +)*** is a set *F* defined with two binary operators (denoted · and + here because of the intuitive laws that exist pertaining to normal 'multiplication' and 'addition'). These laws include:

– ***commutativity*** for both ***(F, ·)*** and ***(F, +)***,
– ***associativity*** for both ***(F, ·)*** and ***(F, +)***,
– ***identity*** for + in ***F*** (denoted ٠),
– ***identity*** for · in ***F* \ *{ ٠}*** (denoted ١),
– ***inverse*** for + in ***F***,
– ***inverse*** for · in ***F* \ *{ ٠}.***
– ***distributivity***,

Definition (٢.٣) [٣]

The characteristic of ***F***, denoted by ***char(F)***, is the smallest positive integer ***p*** such that ***p · a*** = ٠ for all ***a*** ∈ ***F***. If such an integer does not exist, ***char(F)*** is ***zero***.

So ***char(R)*** = ٠, where ***R*** is the field of real number, is an example of a field, and ***char*** $(F_{٢}^{m})$ = ٢, where $F_{٢}^{m}$ is the binary field, is another example of a field.

Definition (٢.٤) [٣٠]

Let ***p*** be a prime number. The finite field F_p, called a ***prime field***, is comprised of the set of integers ***{ ٠, ١, ٢, ٣, …, p*** – ١***}*** with the usual arithmetic modulo ***p***, as follows:

- ***Addition***: if ***a, b*** ∈ F_p, then ***a*** + ***b*** = ***r***, where ***r*** is the remainder when ***a*** + ***b*** is divided by ***p*** and ٠ ≤ ***r*** ≤ ***p*** – ١. This is known as ***addition modulo p***.

- ***Multiplication***: If $a, b \in F_p$, then $a \cdot b = s$, where ***s*** is the remainder when $a \cdot b$ is divided by p and $٠ \leq s \leq p - ١$. This is known as ***multiplication modulo p***.
- ***Inversion***: If a is a non-zero element in F_p , the inverse of a modulo p, denoted $a^{-١}$, is the unique integer $c \in F_p$ for which $a \cdot c = ١$.

Definition (٢.٥)[٣]

a field K is said to be an ***Extension Field*** of F if K contains F.

Definition (٢.٦)[٣]

The ***ring of polynomials*** $F[x]$ in x over a field F is the set of all formal expressions $f(x) = a_٠ + a_١x + ... + a_nx^n$, $n \geq ٠$, $a_i \in F$ for all $i = ٠, ١, ... , n$.

Definition (٢.٧)[٤٧]

Let $f(x) \in F[x]$. if $f(x) \neq ٠$ and $a_n \neq ٠$, then the ***degree*** of $f(x)$, written as $deg\ f(x)$, is n.

Definition (٢.٨)[٣]

A field F is said to be ***algebraically closed*** if for every $f(x) \in F[x]$ such that $deg\ f(x) \geq ١$, $f(x)$ has a root in F.

Definition (٢.٨)[٤]

The field $F_٢^m$, called a ***characteristic two*** or ***binary finite field***, can be viewd as a vector space of dimension m over the field $F_٢$ which consists of the two elements ٠ and ١. That is, there exist m elements $a_٠, a_١, \ldots, a_{m-١}$ in $F_٢^m$ such that each element, a in $F_٢^m$ can be uniquely written in the form : $a = x_٠a_٠ + x_١a_١ + ... + x_{m-١}a_{m-١}$ where $x_i \in \{٠, ١\}$.

Such a set $\{a_٠, a_١, \ldots, a_{m-١}\}$ is called a ***base***[٣] of $F_٢^m$ over $F_٢$. Given such a base a field element a can be representated as the bit string $(a_٠ , a_١, \ldots , a_{m-١})$. There are two kinds of bases: ***Polynomial bases*** and ***normal bases***. And let us organized these as follows:

[٣] A family of vectors $\{v_i\}_{i\in I}$ in a vector space V is a ***base*** iff the following conditions are satisfy :
- $\exists$ a family $\{\delta_i\}_{i\in I}$ of scalars such that finitely many different from ***zero*** and $v = \sum\delta_i v_i$; $\forall\ v\in V$.
- If $\sum\delta_i v_i = ٠$ implies $\delta_i = ٠$ $\forall\ i\in I$.

Polynomial Representation: [٤٨]

The elements of $F_{٢}^{m}$ are polynomials of degree less than m, with coefficients in $F_{٢}$, that is,

$\{a_{m-١}x^{m-١} + a_{m-٢}x^{m-٢} + ... + a_{٢}x^{٢} + a_{١}x + a_{٠} \mid a_{i} \in F_{٢}\}$. These elements can be written in vector form as $(a_{m-١} ... a_{١}\ a_{٠})$. $F_{٢}^{m}$ has ٢^{m} elements. The main operations in $F_{٢}^{m}$ are ***addition*** and ***multiplication***. Some computations involve a polynomial

$f(x) = x^{m} + a_{m-١}x^{m-١} + a_{m-٢}x^{m-٢} + ... + a_{٢}x^{٢} + a_{١}x + a_{٠}$, where $a_{i} \in F_{٢}$. The polynomial $f(x)$ must be irreducible[٤].

- ***Addition:***

  $(a_{m-١} ... a_{١}\ a_{٠}) + (b_{m-١} ... b_{١}\ b_{٠}) = (c_{m-١} ... c_{١}\ c_{٠}) \ni c_{i} = a_{i} + b_{i}$ over $F_{٢}$.

- ***Subtraction:***

  In the field $F_{٢}^{m}$, each element $(a_{m-١} ... a_{١}\ a_{٠})$ is its own additive inverse, since $(a_{m-١} ... a_{١}\ a_{٠}) + (a_{m-١} ... a_{١}\ a_{٠}) = (٠ ... ٠\ ٠)$, where $(٠ ... ٠\ ٠)$ is the additive identity. Thus, addition and subtraction are equivalent operations in $F_{٢}^{m}$.

- ***Multiplication:***

  $(a_{m-١} ... a_{١}\ a_{٠})\ (b_{m-١} ... b_{١}\ b_{٠}) = (r_{m-١} ... r_{١}\ r_{٠})$ where $r_{m-١}x^{m-١} + ... + r_{١}x + r_{٠}$ is the remainder when the polynomial $(a_{m-١}x^{m-١} + ... + a_{١}x + a_{٠})(b_{m-١}x^{m-١} + ... + b_{١}x + b_{٠})$ is divided by the polynomial $f(x)$ over $F_{٢}$. (Note that all polynomial coefficients are reduced modulo ٢).

- ***Exponentiation:***

  The exponentiation $(a_{m-١} ... a_{١}\ a_{٠})^{r}$ is performed by multiplying together ***e*** copies of $(a_{m-١} ... a_{١}\ a_{٠})$.

- ***Multiplicative Inversion:***

  There exists at least one element g in $F_{٢}^{m}$ such that all non-zero elements in $F_{٢}^{m}$ can be expressed as a power of g. Such an element g is called a ***generator*** of $F_{٢}^{m}$. The multiplicative inverse of an element $a = g^{i}$ *is* $a^{-١} = g^{(-i) \bmod (2^{m}-1)}$.

Optimal Normal Basis Representation:

an optimal normal bases (ONB) of $F_{٢}^{m}$ over $F_{٢}$ exists iff one of the following conditions holds [٣٠]:

[٤] $f(x)$ irreducible polynomial which is of degree m means, it cannot be factored into two polynomials over $F_{٢}$, each of degree less than m

١. m + ١ is prime, and ٢ is primitive in $F_{m+١}$. then there is ONB of $F_{٢}^{m}$ over $F_{٢}$, called a Type **I** ONB.

٢. ٢m + ١ is prime, and either
(a) ٢ is primitive in $F_{٢m+١}$, or
(b) ٢m + ١ ≡ ٣(*mod* ٤) and the order of ٢ in $F_{٢m+١}$ is m, then there is ONB of $F_{٢}^{m}$ over $F_{٢}$, called a Type **II** ONB.

Therefore optimal normal basis representations are classified as either Type **I** or Type **II**. So the value of m determines which type shall be used.

now if $F_{٢}^{m}$ only has a type **I** ONB, then let $f(x) = x^{m} + x^{m-١} + ... + x^{٢} + x + ١$.

Otherwise, if $F_{٢}^{m}$ has a type **II** ONB then compute $f(x) = f_{m}(x)$ using the following recursive formulae[٤٩]:

$f_{٠}(x) = ١,$

$f_{١}(x) = x + ١,$

$f_{i+١}(x) = x\,f_{i}(x) + f_{i-١}(x),\ i = ١, ..., m.$

At each stage, the coefficients of the polynomials $f_{i}(x)$ are reduced modulo ٢; hence $f(x)$ is a polynomial of degree m with coefficients in $F_{٢}$.The set of polynomials $\{x, x^{٢}, x^{2^{2}}, ..., x^{2^{m-1}}\}$ forms a basis of $F_{٢}^{m}$ over $F_{٢}$, called a ***normal basis*** [٤٩],[٣٠]. Addition and subtraction are defined as with polynomial representation. Now let us consider the setup for Multiplication:

$f(x) = x^{m} + x^{m-١} + ... + x + ١.$

- ***Construct*** the m by m matrix A whose i^{th} row, i = ٠ ... m – ١, is the bit string corresponding to the polynomial $x^{2^{i}}$ *mod f(x).* (The rows and columns of A are indexed by the integers from ٠ to m – ١.) The entries of A are elements of $F_{٢}$.
- ***Determine*** the inverse matrix $A^{-١}$ of A over $F_{٢}$.
- ***Construct*** the m by m matrix T' whose i^{th} row, i = ٠ ... m – ١, is $x.\ x^{2^{i}}$ *mod f(x)*. Then compute the matrix $T = T'A^{-١}$ over $F_{٢}$.
- ***Determine*** the product terms $l_{i,j}$, for i , j = ٠ ... m – ١, as $l_{i,j} = T(j - i, -i)$. ($T(g, h)$ denotes (g,h)-entry of T with indices reduced modulo m.) Each product term $l_{i,j}$ is an element of $F_{٢}$. It should also be the case that $l_{٠,j}$ = ١ for precisely one j, ٠ ≤ j ≤ m – ١, and that for each i, ٠ ≤ i ≤ m – ١, $l_{i,j}$ = ١ for precisely two distinct j, ٠ ≤ j ≤ m – ١. Hence, only ٢m – ١ of the $m^{٢}$ entries of the matrix T are ١, the rest being ٠. This scarcity of ١'s is the reason that the normal basis is called an ***optimal normal basis***.

This method for representing $F_{٢}^{٤}$ is called an ***optimal normal basis representation.***

Squaring is a very efficient operation when optimal normal basis representation is used. Since exponentiation typically involves many squaring operations, exponentiation is performed far more efficiently using optimal normal basis representation than using polynomial representation.

(٢. ٣) Simple Review of Number Theory

In this section, provides a brief review of fundamental ideas of number theory, to define the ***Legendre Symbol*** that will be used to compute the number of points on ECs, and to give the definition of ***DLP***.

Definition (٢. ١٠) [٤٦]

Let a and b be integers and n is a positive integer . a is ***congruent to b modulo n*** can be denoted by $a \equiv b \ (mod\ n)$ if n is a divisor of $(a - b)$, or equivalently, if $n \setminus (a - b)$. and a is ***not congruent to b modulo n*** can be written as $a \not\equiv b \ (mod\ n)$ if n is not divisor of $(a - b)$

Definition (٢. ١١)[١] , [٤٦], [٥٠]

Let a be integer and n a positive integer, and suppose that $gcd(a,n) = ١$. Then a *i*s called a ***quadratic residue modulo n*** if the congruence $x^{٢} \equiv a \ (mod\ n)$ is solvable (has solution). Otherwise it is called a ***quadratic nonresidue modulo n*** .

Remark (٢. ٢):

Similarly, we can define the ***cubic residues***, and ***fourth-power residues*** ,etc. For example, a is a kth power residue modulo n if the congruence $x^{k} \equiv a \ (mod\ n)$ is solvable. Otherwise ,it is a kth power ***nonresidue modulo n***.

Theorem (٢. ١) (The Chinese Remainder Theorem)

Let $m_{١}, m_{٢}, \dots , m_{r}$ be pairwise relatively prime[٥] positive integers. Then the system of congruences:

[٥] Integers a ,b are called ***relatively prime*** iff $gcd\ (a , b)= ١$

$$\begin{aligned} x &\equiv a_{١}\ (\mathrm{mod}\ m_{١}),\\ x &\equiv a_{٢}\ (\mathrm{mod}\ m_{٢}),\\ &\ \ \vdots\\ x &\equiv a_{r}\ (\mathrm{mod}\ m_{r}), \end{aligned}$$

has a unique solution modulo $M = m_{١}\, m_{٢} \ldots m_{r}$.

Theorem (٢. ٢) (Euler's criterion)

Let p be an odd prime and $gcd(a, p)$ = ١. then a is a ***quadratic residues*** of p iff $a^{(p-١)/٢} \equiv ١ \bmod p$ [٤٦] , [١].

Definition (٢. ١٢) [٤٦]

Let p be an odd prime and a an integer. Suppose that $gcd(a,p)$= ١. Then the ***Legendre symbol***, $\left(\frac{a}{p}\right)$ is defined by

$$\left(\frac{a}{p}\right) = \begin{cases} 1 & \text{if } a \text{ is a quadratic residue modulo } p, \\ -1 & \text{if } a \text{ is a quadratic nonresidue modulo } p. \end{cases}$$

Remark (٢. ٣) [١],[٤٦]

Some elementary properties of the ***Legender symbol***, which can be used to evaluate it, the reader can take it from.

Definition (٢. ١٣) [٤٦]

Let n be an a positive integer and a an integer such that $gcd(a,n)$= ١ . Then the ***order*** of a modulo n, denoted by $ord(a,n)$, is the smallest integer r such that $a^{r} \equiv ١\ (\mathrm{mod}\ n)$.

Definition (٢. ١٤) [٤٦]

Let n be a positive integer, ***Euler's Φ – function,*** denoted by $\Phi(n)$ is defined to be the number of positive integers k less than n which are relatively prime to n .

i.e. $\Phi(n) = \#\{k :\ ١ \leq k \leq n,\ gcd\,(k, n) = ١\}$ or $\Phi(n) = \sum_{\substack{1 \leq k < n \\ gcd(k,n)=1}} 1$

Remark (٢.٤) [١], [٤٦]

We have a fact which says that "***If n is prime number, then every integer $١\le a \le n$ is relatively prime to n***" .So for prime numbers we have the formula: $\Phi(n) = n-١$.

Definition (٢.١٥)[٤٦]

Let n be a positive integer and a is an integer such that $gcd(a,n) = ١$. If the order of an integer a modulo n is $\Phi(n)$, that is, $ord(a,n) = \Phi(n)$, then a is called a ***primitive root*** of n.

(٢.٣.١) Indices

Definition (٢.١٦)[٤٦]

If $x \equiv a \pmod n$, then a is called a ***residue*** of x modulo n . The ***residue class*** of a modulo n, denoted by $[a]_n$, is the set of all those integers that are congruent to a modulo n.

That is, $$[a]_n = \{x : x \in \mathbb{Z} \text{ and } x \equiv a \pmod n \},$$ $$= \{a + kn : k \in \mathbb{Z} \}.$$

Definition (٢.١٧)

The set of all ***residue classes modulo n***, often denoted by $\mathbb{Z}/n\mathbb{Z}$ or $\mathbb{Z}_n$ is the set of all residue class $[a]_n$ where $٠ \le a \le n-١$.

Remark (٢.٥)

One often sees the definition: $\mathbb{Z}/n\mathbb{Z} = \{٠ , ١ , ٢ , ٣ , \ldots , n-١\}$, which should be read as equivalent to $\mathbb{Z}/n\mathbb{Z}$ in the dfintion (٢.١٧) [٤٦].

Remark (٢.٦)

Gauss [٤٦] introduced the concept of ***index*** of an integer modulo n. Given an integer n, if n has primitive root g, then the set $\{g,g^٢,g^٣,\ldots,g^{\Phi(n)}\}$ forms a ***reduced system*** of residues modulo n ; g is a generator of the cyclic group which is the reduced residues modulo n set. Hence, if $gcd(a , n) = ١$, then a can be expressed in the form :

$$a \equiv g^k \pmod n$$

For a suitable k with $١\le k \le \Phi(n)$. This motivates our following definition, which is an analogue of the real base ***logarithm function***.

Definition (٢. ١٨)[٤٦]

Let g be a primitive root of n .If $gcd(a, n)=$ ١, then the smallest positive integer k such that $a \equiv g^k \ (mod\ n)$ is called the ***index*** of a to the base g modulo n and is denoted by $ind_g\ a$.

Remark (٢. ٧)[٤٦]

We have by remark (٢.٦) $a \equiv g^{ind_g a} \ (mod\ n).$
The function $ind_g a$ is sometimes called the ***discrete logarithm*** and is denoted by $log_g a$ so that

$$a \equiv g^{log_g a} \ (mod\ n).$$

Generally, the ***discrete logarithm*** is a computationally intractable problem; no efficient algorithm has been found for computing ***discrete logarithms*** and hence it has important applications in public key cryptography.

Theorem (٢. ٣) (*Index theorem*):

If g is a primitive root modulo n, then $g^x \equiv g^y \ (mod\ n)$ iff $x \equiv y \ (mod\ \Phi(n)).$

Remark (٢ .٨)[٤٦]

The ***DLP*** can be described as follows:

$$\left.\begin{array}{ll} \text{Input:} & a, b, n \in N \\ \text{Output:} & x \in N \text{ with } a^x \equiv b \ (mod\ n\ ) \\ & \text{if such an } x \text{ exists,} \end{array}\right\}$$

Where the modulus n can either be a composite or a prime [٤٦].

There are essentially three different categories of algorithms in use for computing discrete logarithm:

- Algorithms that work for arbitrary group.
- Algorithms that work well in finite group, for which the order of the groups has no large prime factors; more specifically, algorithms that work for groups with smooth orders[١].
- Algorithms that exploit methods for representing group elements as products of elements from a relatively small set.

[١] A positive integer is called ***smooth*** if it has no large prime factors; it is called ***y-smooth*** if it has no large prime factors exceeding y .

These algorithms computes the ***DLP*** in general, the reader can take more information from [٤٦].

(٢. ٣. ٢) Fast Modular Exponentiations

In this subsection, we describe the important problem which is said that "***what is the result of** x^r **mod n , when r is large*** [٧]" .

The ***method of repeated squaring*** will solve this problem efficiently using the binary representation of ***r*** [٤٦]. This method can be illustrated by the following: Suppose we wish to compute $a^{١٠٠}$ for any integer ***a***, then to compute it with the above method, at first must be written ١٠٠ as binary representation .

So $100_{10} = 1100100_2 := e_6\, e_5\, e_4\, e_3\, e_2\, e_1\, e_0$, then compute

$$a^{100} = ((((((a)^2.a)^2)^2)^2.a)^2)^2$$

$$\Rightarrow a,\ a^3,\ a^6,\ a^{12},\ a^{24},\ a^{25},\ a^{50},\ a^{100}\ .$$

Note that for each e_i , if e_i = ١, we perform a squaring and a multiplication operation (except $e_٦$ = ١ , for which we just write down ***a***, as indicated in the first bracket), otherwise, we perform only a squaring operation. That is,

e_6	*1*	a	*initialization*
e_5	*1*	$(a)^2.a$	*squaring and multiplication*
e_4	*0*	$((a)^2.a)^2$	*squaring*
e_3	*0*	$(((a)^2.a)^2)^2$	*squaring*
e_2	*1*	$((((a)^2.a)^2)^2)^2.a$	*squaring and multiplication*
e_1	*0*	$(((((a)^2.a)^2)^2)^2.a)^2$	*squaring*
e_0	*0*	$\underbrace{((((((a)^2.a)^2)^2)^2.a)^2)^2}_{\substack{\Downarrow \\ a^{100}}}$	*squaring*

[٧] x^r ***mod n***, called modular exponentiation . In addition, the algorithm, which solves it, is called ***fast modular exponentiations.***

(٢. ٤) Projective Coordinates

From now on, F will always denote a field, unless otherwise stated.

Definition (٢. ١٩)

The ***affine plane*** $A^{٢}$ is the usual plane, $A^{٢}(F) = \{(x, y) / x, y \in F\}.$

Definition (٢. ٢٠)[١٣]

The ***projective space*** is the set of equivalence classes of tuples $(X_{٠}, X_{١}, \dots , X_n)$ (not all components ***zero***) where two tuples are said to be ***equivalent*** if they are scalar multiples of one another, i.e.

$P^n(F) = \{(X_{٠}, X_{١}, \dots, X_n) - (٠, ٠, \dots, ٠) / (\lambda X_{٠}, \lambda X_{١}, \dots, \lambda X_n) \sim (X_{٠} , X_{١}, \dots,X_n),\lambda \in Z \setminus \{٠\}.\}$

Then each equivalence class $(X_{٠}, X_{١}, \dots, X_n)$ is called ***projective point*** and $X_{٠}, X_{١}, \dots, Xn$ are called the ***homogeneous coordinates*** of that point [٨].

Definition (٢. ٢١)[٥١]

The ***projective plane*** $P^{٢}(F)$ over F is

$$P^{٢}(k) = \{(X, Y, Z) \in F^{٣} \mid (X, Y, Z) \neq (٠, ٠, ٠)\} /\sim$$

Where $(X, Y, Z) \sim (X', Y', Z')$ iff there exists a $\lambda \neq ٠$ such that $(X', Y', Z') = (\lambda X, \lambda Y, \lambda Z)$.

Definition (٢. ٢٢)

A ***direction*** is an equivalent class of parallel line in $A^{٢}(F)$**,** that is, a collection of all lines parallel to given line in $A^{٢}(F)$**.**

The set of all direction can be described as the set of all lines in $A^{٢}(F)$ passing through the origin, since every line in $A^{٢}(F)$ is parallel to a unique line passing through the origin. Thus, the set of all

[٨] The relation $\sim$ in a set R is equivalence relation in R, iff the following condition are hold [٤٧]:

- $a \sim a$, for all $a \in R$,
- If $a \sim b$ for some $a, b \in R$, then $b \sim a$,
- If $a \sim b$ and $b \sim c$ for some $a, b, c \in R$, then $a \sim c$.

The set $(a) = \{x \in R / x \sim a\}$ denoted to the equivalence class .

In the usual way we use the symbol [ ] to refer the equivalence class, but in this thesis, we use this symbol to refer the references, therefore we use ( ) to refer the equivalence class.

directions in $A^{٢}(F)$ is described by the points (A, B) of the projective line $P^{١}(F)$, where each (A, B) corresponds to the line $Ay = Bx$. For A, B not both zero, $(A', B') \sim (A, B)$ iff $(\lambda A', \lambda B') \sim (A, B)$ for some $\lambda \neq ٠ \in F$.

Definition (٢. ٢٣)

The ***projection line*** $P^{١}(F)$ over F is the set of equivalence classes (A, B), that is, the set of all directions in $A^{٢}(F)$.

As such, the projective plane $P^{٢}(F)$ can also be defined as

$$P^{٢}(F) = A^{٢}(F) \cup P^{١}(F).$$

Definition (٢. ٢٤)

The set of " ***extra points*** " in $P^{٢}(F)$ associated to the directions in $A^{٢}(F)$ are called ***points at infinity***, denoted by L_{∞} called the ***line at infinity***.

The following maps show how to identify the definitions of the projection plane:

$$\{(X, Y, Z) \in F^{٣} \mid (X, Y, Z) \neq (٠, ٠, ٠)\}/\sim \leftrightarrow A^{٢}(F) \cup P^{١}(F).$$

For each projective point (X, Y, Z) in $P^{٢}(F)$ with $Z \neq ٠$, we can " divide throughout " by Z to yield $(x, y, ١)$ where $x = X/Z$ and $y = Y / Z$. this process is called ***dehomogenization*** with respect to Z (the reverse process is called ***homogenization***). Then there is a one to one correspondence between the point $(x, y, ١) \in P^{٢}(F)$ and the point $(x,y) \in A^{٢}(F)$. In the other words, for $Z \neq ٠$,

$$(X,Y,Z) \mapsto \left(\frac{X}{Z}, \frac{Y}{Z}\right) \mapsto \left(\frac{X}{Z}, \frac{Y}{Z}, 1\right) = (X,Y,Z)$$

The remaining points with $Z = ٠$, the points at infinity, then corresponds to a copy of $P^{١}(F)$ [٣].

(٢. ٥) Arithmetical Operation of Elliptic Curves

The mathematics associated with EC is old; formerly "pure" with no practical applications the math is very deep and fascinating. ECs as algebraic (geometric) entities have been studied extensively for the ١٥٠ years, and from these studies has emerged a rich and deep theories [٤],[١٦]. In this section the important mathematical operation on EC which is needed for researcher in ECCg.

Let us start with the fact which say that (*An Elliptic Is Not An Ellipse!*)[٩]

An elliptic is not an ellipse [٤٦], an EC is a type of cubic curve, While ellipse is a degree ٢ equation with the standard equation for ellipses centered at the origin [٥٢] :

– ***Foci*** on x-axis : $\frac{x^2}{a^2}+\frac{y^2}{b^2}=1 \quad (a>b)$

Center-to-***focus*** distance: $c=\sqrt{a^2-b^2}$

Foci: $(\pm c,0)$

Vertices: $(\pm a,0)$.

– ***Foci*** on y-axis : $\frac{x^2}{b^2}+\frac{y^2}{a^2}=1 \quad (a>b)$

Center-to-***focus*** distance: $c=\sqrt{a^2-b^2}$

Foci: $(0,\pm c)$

Vertices: $(0,\pm a)$.

(However, given such an ellipse, you could try to compute the arc length of a certain portion of the curve; the integral, which arises, can be associated to an EC [٥٢].) To study an EC we must start with " ***Weierstrass Equation*** " .

Definition (٢. ٢٥)[٣٠]

A ***Weierstrass equation*** is a homogeneous equation [١٠] of degree ٣ and has the form

$Y^{٢}Z + a_{١}XYZ + a_{٣}YZ^{٢} = X^{٣} + a_{٢}X^{٢}Z + a_{٤}XZ^{٢} + a_{٦}Z^{٣}$… … … … … … ..(١)

where $a_{١}, a_{٢}, a_{٣}, a_{٤}, a_{٦} \in K$, where K is any field.

The ordering of subscripts of the coefficients in (١) have these ways because of the following ((it is often said that "X has degree ٢" and "Y has degree ٣" because of the following: for large X, the curve extends to infinity much like the function $Y = X^{٣/٢}$, which can be parameterized by $X = T^{٢}$ and $Y = T^{٣}$. The subscripts of the coefficients in (١) therefore indicate the degrees that must be given to the

[٩] ***An ellipse*** is the set of points in a plane whose distance from two fixed point in the plane have a constant sum [٥٢].

[١٠] ***Homogeneous*** means every degree in this equation are equal .

coefficients in order that the equations are homogeneous; that is, each term has a total degree of ٦ [٣], [٤١])) .

Remark (٢. ٩)

"***The Weierstrass equation may be singular or may be non singular***". To explain this statement we must know the following [٣٠] :

- A point p is simple in W **(**where W has a form: $W = Y^{٢}Z + a_{١}XYZ + a_{٣}YZ^{٢} - X^{٣} - a_{٢}X^{٢}Z - a_{٤}XZ^{٢} - a_{٦}Z^{٣} = ٠$ **)** if at least one of the partial derivatives $\frac{\partial W}{\partial X}, \frac{\partial W}{\partial Y}, \frac{\partial W}{\partial Z}$ is non zero at p , other wise p is called ***singular***.
- If W has one singular point then we say that W is ***singular*** , other wise W is ***non–singular*** (***smooth***) .

Examples (٢. ١)

- $S_{١} : Y^{٢} + X^{٣} + X = ٠ ,$

since $\frac{\partial S_1}{\partial X} = ٣X^{٢} + ١ \neq ٠ \quad \forall X,$ then $S_{١}$ is non singular (smooth).

- $S_{٢} : Y^{٢} - X^{٣} - X^{٢} = ٠ ,$

since $\frac{\partial S_2}{\partial X} = -٣X^{٢} - ٢X$, and $\frac{\partial S_2}{\partial Y} = ٢Y ,$

and since $\frac{\partial S_2}{\partial X}(٠,٠) = \frac{\partial S_2}{\partial Y}(٠,٠) = ٠$, then $S_{٢}$ has singular point $(٠,٠)$, so $S_{٢}$ is singular .

Definition (٢. ٢٦)[٥٣]

An ***EC*** E is the set of all solutions in $P^{٢}(K)$ of a smooth Weierstrass equation. There is exactly one point in E with $Z-$coordinate is equal to ٠, namely $(٠, ١, ٠)$. This point is the point at infinity [١١] and it is denoted by O .

For convenience, the Weierstrass equation for ECs can be written using non-homogeneous coordinates [٣٠],

[١١] if we substituting $Z = ٠$,in the Weierstrass equation, we have $X^{٣} = ٠$, which implies $X = ٠$. the only equivalence class of triples with $X = ٠ = Z$ is the class of $(٠, ١, ٠)$ [٣].

where $x = \frac{X}{Z}, y = \frac{Y}{Z}$,

$y^{٢} + a_{١}xy + a_{٣}y = x^{٣} + a_{٢}x^{٢} + a_{٤}x + a_{٦}$ … … … … … … … … … … .(٢)

An EC E is then the set of solutions to equation above in the affine plane $A^{٢}(K) = K \times K$, together with the extra point at infinity O.

Definition (٢. ٢٧)

A rational point on an EC E over F is called a ***F–point***. Let $E(F)$ denoted the set of ***F–point(x,y)∈F***^{٢} that satisfies equation (٢)***,*** along with O. if K is any extension field of F, then $E(F)$ denotes the set of ***K–point (x, y)*** $\in K^{٢}$ that satisfies equation (٢)***,*** along with O.

Here, we define an important quantity related to E, namely, the ***discriminant*** and ***j–invariant*** of E. Its original form in the general case may look very complicated, but in specific cases, it will look much simpler. We uses this concept in chapter three for determining the types of EC.

Definition (٢. ٢٨)

Δ is called the ***discrimiant*** of E and $j(E)$ is called the ***j– invariant*** of E are given by :

$$\Delta = -b_2{}^2 b_8 - 8b_4{}^3 - 27b_6{}^2 + 9b_2b_4b_6$$

$$j(E) = \frac{c_4{}^3}{\Delta}$$

Where

$$b_2 = a_1{}^2 + 4a_2$$

$$b_4 = 2a_4 + a_1a_3$$

$$b_6 = a_3{}^2 + 4a_6$$

$$b_8 = a_1{}^2a_6 + 4a_2a_6 - a_1a_3a_4 + a_2a_3{}^2 - a_4{}^2$$

$$c_4 = b_2{}^2 - 24b_4.$$

Next, we look at the possible different forms of the equation of an EC E under different characteristics of the defining field F. The objective is to get a simpler form of the defining equation. We split it into ٣ cases: ***char(F)*** ≠ ٢ or ٣, ***char(F)*** = ٣ and ***char(F)*** = ٢.

– ***char(F)* ≠ ٢ or ٣** [٣]:

if ***char(F)*** ≠ ٢, complete the square on the left hand side of equation (٢) and the corresponding admissible change of variables $(x,y) \mapsto (x, y - \frac{1}{2}(a_1x + a_3))$ Transforms ***E*** into

$$y^2 = x^3 + a_2'\, x^2 + a_4'x + a_6' \quad \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \text{(٣)}$$

Further, if ***char(F)*** ≠ ٣, a similar process can be carried out on the right hand side of equation (٢).

The transformation $(x,y) \mapsto (x - \frac{1}{3}a_2', y)$ yields from equation (٣) is

$$y^2 = x^3 + a_4''x + a_6'' \quad \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \text{(٤)}$$

– ***char(F)* = ٣:**

if $a_2' \neq 0$, the transformation $(x,y) \mapsto (x + \frac{a_4'}{a_2'}, y)$ yields from equation (٣) is

$$y^2 = x^3 + a_2''\, x^2 + a_6'' \quad \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \text{(٥)}$$

if $a_2' = 0$, then from equation (٣), we immediately have the desired form

$$y^2 = x^3 + a_4'x + a_6' \quad \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \text{(٦)}$$

– ***char(F)* = ٢:**

(١) the supersingular case, ***a*** ١ = ٠**:**

The substitution $(x,y) \mapsto (x + a_2, y)$ in equation (٢) gives

$$y^2 + a_3'y = x^3 + a_4'x + a_6' \quad \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \text{(٧)}$$

(٢) The nonsupersingular case, ***a*** ١ ≠ ٠**:**

The substitution

$(x,y) \mapsto (a_1{}^2x + \frac{a_3}{a_1}, a_1{}^3y + \frac{a_1{}^2a_4 + a_3{}^2}{a_1{}^3})$ in equation (٢) gives

$$y^2 + xy = x^3 + a_2'\, x^2 + a_6' \quad \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \text{(٨)}$$

(٢. ٥. ١) Group Low of an Elliptic Curve

Many cryptosystems often require the use of algebraic groups. ECs may be used to form EC groups. A group is a set of elements with custom-defined arithmetic operations on those elements. For EC groups, these specific operations are defined geometrically. By introducing more stringent properties to the elements of a group, such as limiting the number of points on such a curve creates an underlying field for an EC group. ECs are first examined over real numbers in order to illustrate the geometrical properties of EC groups. Thereafter, ECs groups are examined with the underlying fields of F_p (where p is a prime) and $F_{٢^m}$ (a binary representation with ٢^m elements)[١٢].

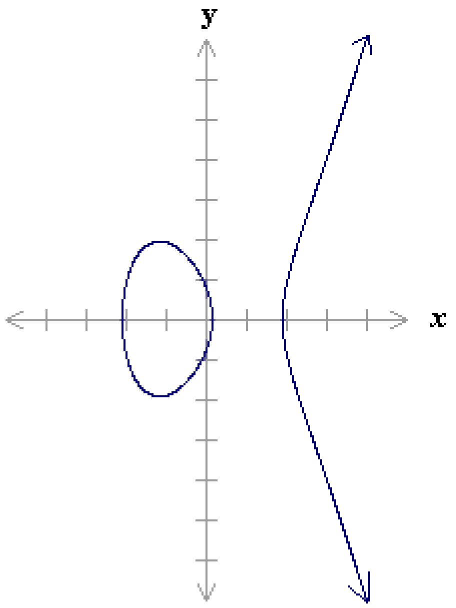
(٢.٥.١.١) Elliptic Curve Groups over Real Numbers

An EC over real numbers may be defined as the set of points (x, y) which satisfy an EC equation of the form: $y^٢ = x^٣ + ax + b$, where x, y, a and b are real numbers. Each choice of the numbers a and b yields a different EC.

Example (٢.٢)

Let $y^٢ = x^٣ – ٤x + ٠.٦٧$, then the graph of this curve is shown below:

[١٢] the reader can see the appendix C for further discussion of formulas law of operation on EC.



$y^2 = x^3 - 4x + 0.67$

If $x^٣ + ax + b$ contains no repeated factors, or equivalently if $٤a^٣ + ٢٧b^٢$ is not ٠, then the EC $y^٢ = x^٣ + ax + b$ can be used to form a group. An EC group over real numbers consists of the points on the corresponding EC, together with a special point O "***point at infinity*** " [٤].

Elliptic Curve Addition: A Geometric Approach

EC groups are additive groups; that is, their basic function is addition. The addition of two points in an EC is defined geometrically.

The negative of a point $P = (x_P , y_P)$ is its reflection in the x-axis: the point $-P$ is $(x_P ,-y_P)$. Notice that for each point P on an EC, the point $-P$ is also on the curve.

Definition (Adding distinct two points) (٢. ٢٩)

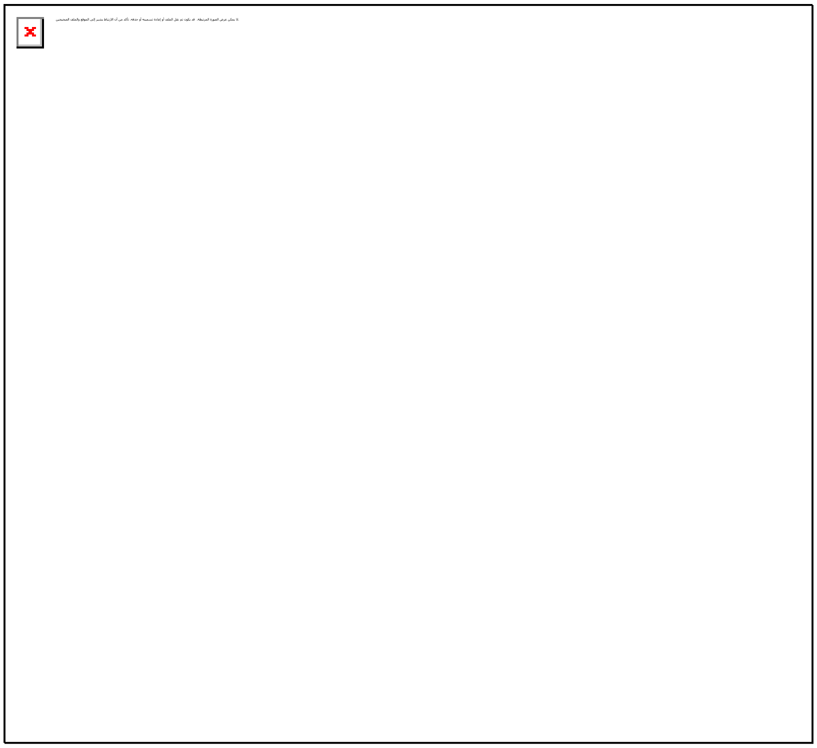
Suppose that P and Q are two distinct points on an EC, and the P is not $-Q$. To add the points P and Q, a line is drawn through the two points. This line will intersect the EC in exactly one more point, call –

R. The point – ***R*** is reflected in the ***x***-axis to the point ***R***. The law for addition in an EC group is ***P*** + ***Q*** = ***R.***

Example (٢.٣)

By the following graphing we will compute ***P(–٢.٣٥, –١.٨٦)*** + ***Q(–٠.١, ٠.٨٣٦).***

Solution

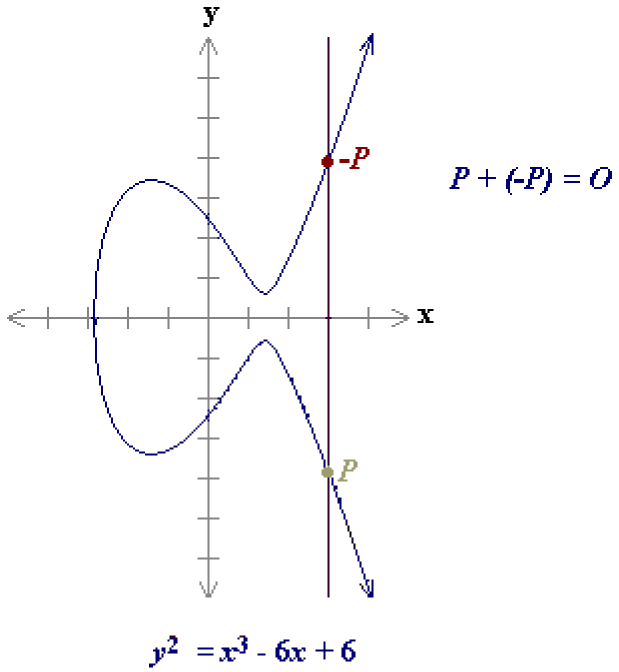

لا يمكن عرض الصورة المرتبطة. قد يكون تم نقل الملف أو إعادة تسميته أو حذفه. تأكد من أن الارتباط يشير إلى الموقع والملف الصحيحين.

Definition (Adding the points P and – P) (٢.٣٠)

The line through ***P*** and – ***P*** is a vertical line, which does not intersect the EC at a third point; thus the points ***P*** and – ***P***, cannot be added as previously. It is for this reason that the EC group includes the point at infinity ***O***. By definition of adding distinct two points, ***P*** + ***(– P)*** = ***O***. Because of this equation, ***P*** + ***O*** = ***P*** in the EC group. ***O*** is called the additive identity of the EC group; all ECs have an additive identity.

Example (٢.٤)

The following graph shows the ***P*** + ***(– P)*** for any point on ***E:*** $y^{٢} = x^{٣} - ٦x + ٦$.



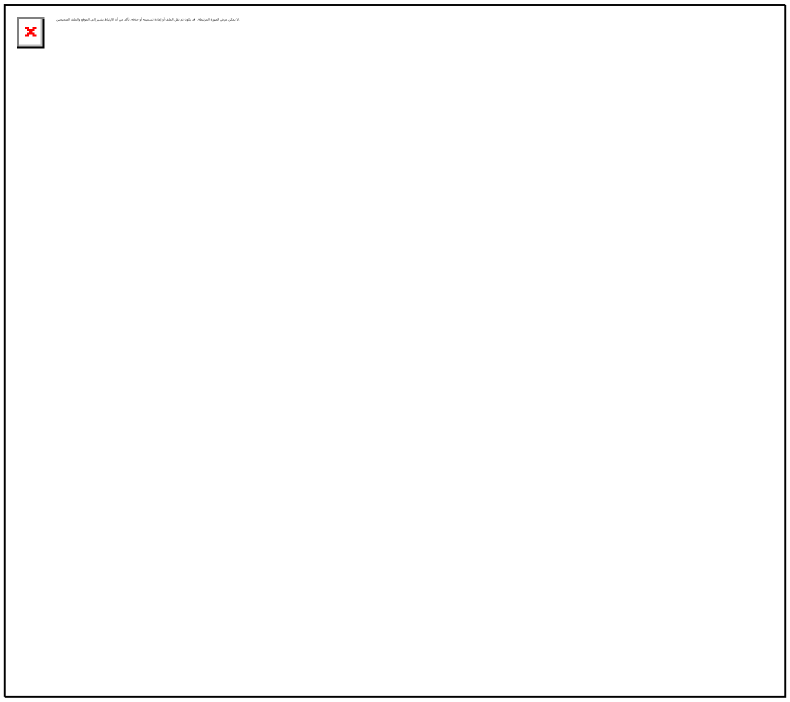
Definition (Doubling the point P) (٢.٣١)

To add a point ***P*** to itself, a tangent line to the curve is drawn at the point ***P***. If y_P is not ٠, then the tangent line intersects the EC at exactly one other point, ***– R. – R*** is reflected in the ***x***-axis to ***R.*** This operation is called ***doubling the point P***; the law for doubling a point on an EC group is defined by: ***P*** + ***P*** = ***٢P*** = ***R.***

Example (٢.٥)

The following graph shows the ***٢P,*** where ***P*** = ***(٢, ٢.٢٦٥)*** .

Solution

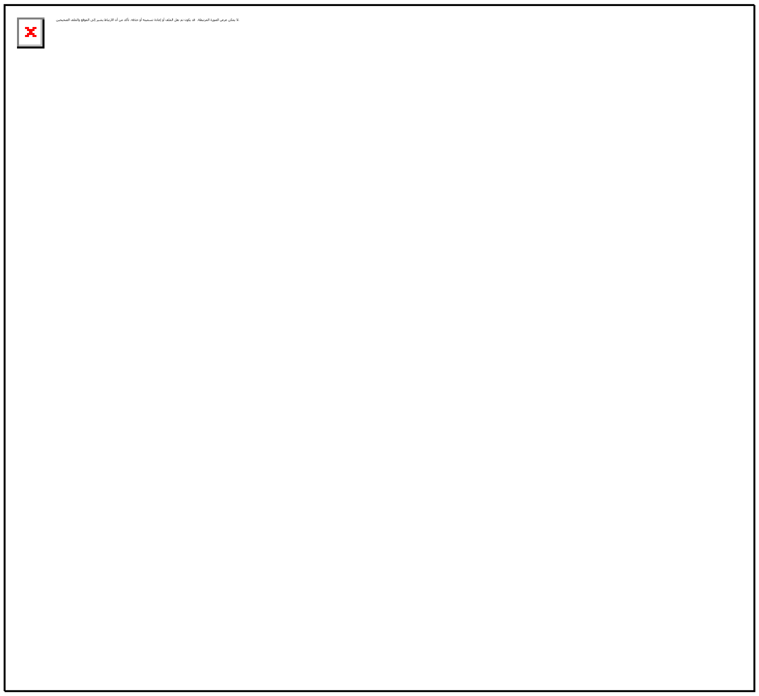

لا يمكن عرض الصورة المرتبطة. قد يكون تم نقل الملف أو إعادة تسميته أو حذفه. تأكد من أن الارتباط يشير إلى الموقع والملف الصحيحين.

Now, If a point ***P*** is such that y_P = ٠, then the tangent line to the EC at ***P*** is vertical and does not intersect the EC at any other point. By definition of adding distinct two points, then ***٢P*** = ***O*** for such a point ***P***.

If ***٣P*** has to be found in this situation, one can add ***٢P*** + ***P***. This becomes ***P*** + ***O*** = ***P***, Thus ***٣P = P, ٤P = O, ٥P = P, ٦P = O, ٧P =P*** , etc.

Example (٢.٦)

The following graph shows the ***٢P*** , where ***P = (١.١, ٠).***


لا يمكن عرض الصورة المرتبطة. قد يكون تم نقل الملف أو إعادة تسميته أو حذفه. تأكد من أن الارتباط يشير إلى الموقع والملف الصحيحين.

Elliptic Curve Addition: An Algebraic Approach

Although the previous geometric descriptions of ECs provide an excellent method of illustrating EC arithmetic, it is not a practical way to implement arithmetic computations. Algebraic formulae are constructed to efficiently compute the geometric arithmetic.

Definition (Adding distinct points) (٢.٣٢)

When $\boldsymbol{P} = (\boldsymbol{x_P}, \boldsymbol{y_P})$ and $\boldsymbol{Q} = (\boldsymbol{x_Q}, \boldsymbol{y_Q})$ are not negative of each other,

$\boldsymbol{P} + \boldsymbol{Q} = \boldsymbol{R}$, where $\lambda = \dfrac{y_P - y_Q}{x_P - x_Q}$,

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

$$y_R = -y_P + \lambda(x_P - x_R),$$

Note that λ is the slope of the line through $\boldsymbol{P}$ and $\boldsymbol{Q}$.

Definition (Doubling the point P) (٢.٣٣)

When $\boldsymbol{y_P}$ is not ٠, ٢$\boldsymbol{P} = \boldsymbol{R}$ where

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

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

$$y_R = -y_P + \lambda(x_P - x_R),$$

Recall that a is one of the parameters chosen with the EC and that λ is the tangent on the point P.

(٢.٥.١.٢) Elliptic Curve Groups over F_p

Calculations over the real numbers are slow and inaccurate due to round-off error. Cryptographic applications require fast and precise arithmetic; thus, EC groups over the finite fields of F_p and $F_{٢}^{m}$ are used in practice.

An EC with the underlying field of F_p can be formed by choosing the variables a and b within the field of F_p. The EC includes all points (x , y) which satisfy the EC equation modulo p (where x and y are numbers in F_p).for example**:** $y^{٢}$ $mod\ p$ = $x^{٣}$ + ax + b $mod\ p$ has an underlying field of F_p if a and b are in F_p. so the equation of EC over F_p can be written as follows:

$$y^{٢} = x^{٣} + ax + b \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \quad \ldots \ldots \ldots (٩)$$

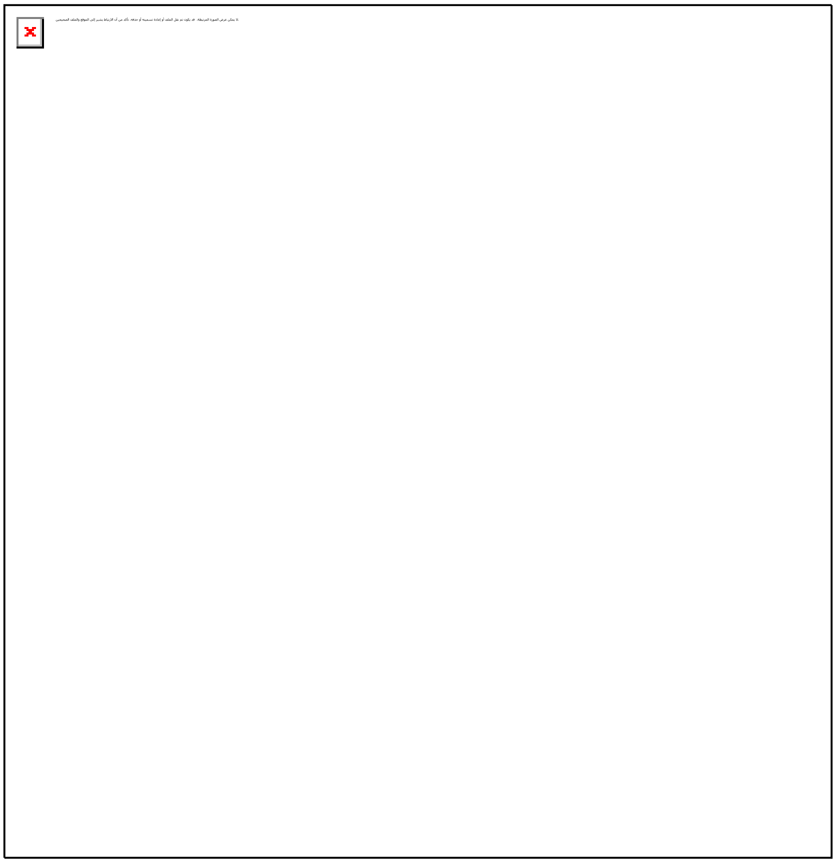
where $a , b \in F_p$.

If $x^{٣}$ + ax + b contains no repeating factors (or, equivalently, if $٤a^{٣}$ + $٢٧b^{٢}$ $mod\ p$ is not ٠), then the EC can be used to form a group. An EC group over F_p consists of the points on the corresponding EC, together with a special point O called the ***point at infinity***. There are finitely many points on such an EC.

Example (٢.٧)

Consider an EC over the field $F_{٢٣}$. With a = ١ and b = ٠, the EC equation is $y^{٢} = x^{٣} + x$.

The ٢٣ points, which satisfy this equation, are: ***(٠,٠) (١,٥) (١,١٨) (٩,٥) (٩,١٨) (١١,١٠) (١١,١٣) (١٣,٥) (١٣,١٨) (١٥,٣) (١٥,٢٠) (١٦,٨) (١٦,١٥) (١٧,١٠) (١٧,١٣) (١٨,١٠) (١٨,١٣) (١٩,١) (١٩,٢٢) (٢٠,٤) (٢٠,١٩) (٢١,٦) (٢١,١٧)*** **.** These points may be graphed as below:


لا يمكن عرض الصورة المرتبطة. قد يكون تم نقل الملف أو إعادة تسميته أو حذفه. تأكد من أن الارتباط يشير إلى الموقع والملف الصحيحين.

Note that there is ***two*** points for every x value. Even though the graph seems random, there is still symmetry about y = ١١.٥. Recall that ECs over real numbers, there exists a negative point for each point, which is reflected through the x-axis. Over the field of $F_{٢٣}$, the negative components in the y-values are taken modulo ٢٣, resulting in a positive number as a difference from ٢٣.

Hence $-P = (x_P, (-y_P \bmod ٢٣))$.

Note that these rules are the same as those for EC groups over real numbers, with the exception that computations are performed modulo p.

Arithmetic in an Elliptic Curve Group over F_p

There are several major differences between EC groups over F_p and over real numbers. EC groups over F_p have a finite number of points, which is a desirable property for cryptographic purposes. Since these curves consist of a few discrete points, it is not clear how to “connect the dots” to make their graph look like a curve. It is not clear how geometric relationships can be applied. As a result, the geometry used in EC groups over real numbers cannot be used for EC groups over F_p. However, the algebraic rules for the arithmetic can be adapted for ECs over F_p. Unlike ECs over real numbers, computations over the field of F_p involve no round off error - an essential property required for a cryptosystem.

Definition (Adding distinct points in F_p) (٢.٣٤)

The negative of the point $P = (x_P, y_P)$ is the point $-P = (x_P, -y_P \bmod p)$. If P and Q are distinct points such that P is not $-Q$, then $P + Q = R$, where

$$\lambda = \frac{y_P - y_Q}{x_P - x_Q} \bmod p,$$

$$x_R = (\lambda^2 - x_P - x_Q) \bmod p,$$

$$y_R = (-y_P + \lambda(x_P - x_R)) \bmod p,$$

Note that λ is the slope of the line through P and Q.

Definition (Doubling the point in F_p) (٢.٣٥)

When y_P is not ٠, ٢$P = R$ where

$$\lambda = \frac{3x_P^2 + a}{2y_P} \bmod p,$$

$$x_R = (\lambda^2 - 2x_P) \bmod p,$$

$$y_R = (-y_P + \lambda(x_P - x_R)) \bmod p,$$

Recall that a is one of the parameters chosen with the EC and that λ is the slope of the line through P and Q .

Example (٢.٨)

(١) In the EC group defined by $y^{٢} = x^{٣} + x +$ ٧ over $F_{١٧}$, we will compute $P + Q$ if $P =$ (٢, ٠) and $Q =$ (١, ٣).

Solution

$\lambda = (y_P - y_Q) / (x_P - x_Q) \bmod p =$ (– ٣) / ١ mod ١٧ = – ٣ mod ١٧ = ١٤

$x_R = (\lambda^{٢} - x_P - x_Q) \bmod p =$ (١٩٦ – ٢ – ١) mod ١٧ = ١٩٣ mod ١٧ = ٦

$y_R = (-y_P + \lambda(x_P - x_R)) \bmod p =$ (٠ + ١٤٠ (٢ – ٦)) mod ١٧ = – ٥٦ mod ١٧ = ١٢

Thus $P + Q =$ (٦, ١٢).

(٢) In the EC group defined by $y^{٢} = x^{٣} + x + ٧$ over $F_{١٧}$, we will compute $٢P$ if $P = (١, ٣)$.

Solution

$\lambda = (٣x_P{}^{٢} + a) / (٢y_P) \bmod p = (٣ + ١) \cdot ٦– ١ \bmod ١٧ = ٤ \cdot ٣ \bmod ١٧ = ١٢$

$x_R = (\lambda^{٢} – ٢x_P \bmod p) = (١٤٤ – ٢ \bmod ١٧) = ١٤٢ \bmod ١٧ = ٦$

$y_R = (–y_P + \lambda\, (x_P – x_R)) \bmod p = (– ٣ + ١٢ \cdot (١ – ٦)) \bmod ١٧ = – ٦٣ \bmod ١٧ = ٥$

Thus $٢P = (٦, ٥)$.

(٢.٥.١.٣) Elliptic Curve Groups over Binary Field

A non-supersingular [١٣] EC $E(F_{٢}^{m})$ over $F_{٢}^{m}$ defined by the parameters $a, b \in F_{٢}^{m}$, is the set of solutions $(x, y) \in F_{٢}^{m} \times F_{٢}^{m}$ to the equation $y^{٢} + xy = x^{٣} + ax^{٢} + b$ [٥٤].

An EC group over $F_{٢}^{m}$ consists of the number of points in $E(F_{٢}^{m})$ is denoted by $\# E(F_{٢}^{m})$, compute as follows:

$$q + ١ – ٢\sqrt{q} \le \# E(F_{٢}^{m}) \le q + ١ + ٢\sqrt{q}$$

where $q = ٢^{m}$, together with a point at infinity, O.

Example (٢.٩)

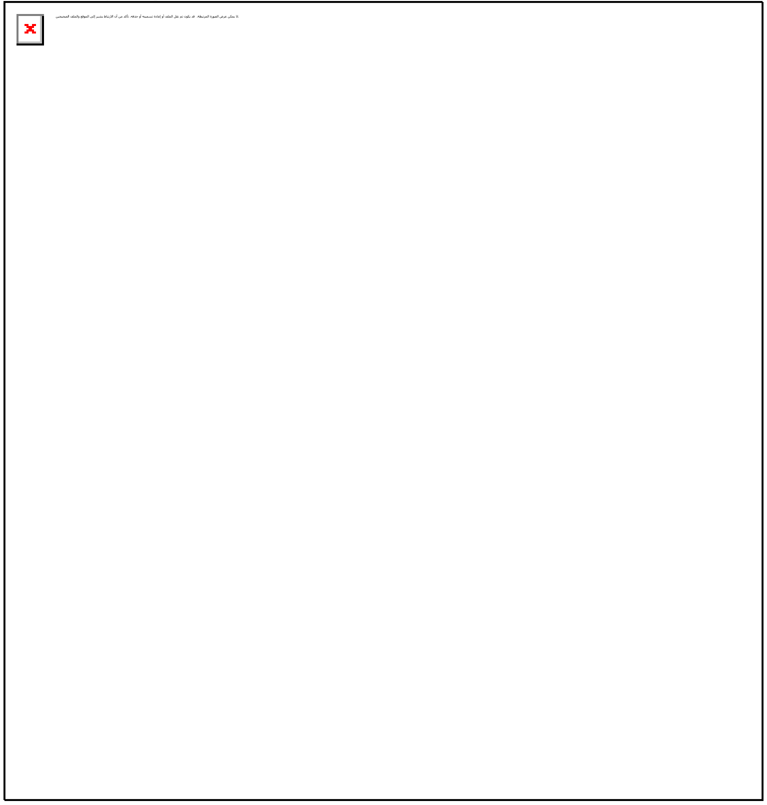
Consider the field $F_{٢}{}^{٤}$, defined by using polynomial representation with the irreducible polynomial $f(x) = x^{٤} + x + ١$. The element $g = (٠٠١٠)$ is a generator for the field. The powers of g are:

$g^{٠} = (٠٠٠١)$, $g^{١} = (٠٠١٠)$, $g^{٢} = (٠١٠٠)$, $g^{٣} = (١٠٠٠)$, $g^{٤} = (٠٠١١)$, $g^{٥} = (٠١١٠)$, $g^{٦} = (١١٠٠)$, $g^{٧} = (١٠١١)$, $g^{٨} = (٠١٠١)$, $g^{٩} = (١٠١٠)$, $g^{١٠} = (٠١١١)$, $g^{١١} = (١١١٠)$, $g^{١٢} = (١١١١)$, $g^{١٣} = (١١٠١)$, $g^{١٤} = (١٠٠١)$, $g^{١٥} = (٠٠٠١)$.

In a true cryptographic application, the parameter m must be large enough to preclude the efficient generation of such a table otherwise the cryptosystem can not be considered as asecure. In today's practice, $m = ١٦٠$ is a suitable choice. The table allows the use of generator notation (g^{r}) rather than bit string notation, as shown in the following illustration, In addition, using generator notation allows multiplication without reference to the irreducible polynomial $f(x) = x^{٤} + x + ١$. Consider the EC $y^{٢} + xy = x^{٣} + g^{٤}x^{٢} + ١$. Here $a = g^{٤}$ and $b = g^{٠} = ١$.

[١٣] We will defined it in the chapter three.

Then the fifteen points, which satisfy this equation, are:
$(١, g^{١٣}), (g^{٣}, g^{١٣}), (g^{٥}, g^{١١}), (g^{٦}, g^{١٤}), (g^{٩}, g^{١٣}), (g^{١٠}, g^{٨}), (g^{١٢}, g^{١٢}),$
$(١, g^{٦}), (g^{٣}, g^{٨}), (g^{٥}, g^{٣}), (g^{٦}, g^{٨}), (g^{٩}, g^{١٠}), (g^{١٠}, g), (g^{١٢}, ٠), (٠, ١).$
These points are graphed below:



لا يمكن عرض الصورة المرتبطة. قد يكون تم نقل الملف أو إعادة تسميته أو حذفه. تأكد من أن الارتباط يشير إلى الموقع والملف الصحيحين.

Arithmetic in an Elliptic Curve Group over $F_{٢^m}$

EC groups over $F_{٢^m}$ have a finite number of points, and their arithmetic involves no round off error. This combined with the binary nature of the field, $F_{٢^m}$ arithmetic can be performed very efficiently by a computer. The following algebraic rules are applied for arithmetic over $F_{٢^m}$:

Definition (Adding distinct points in $F_{٢^m}$) (٢.٣٦)

The negative of the point $P = (x_P, y_P)$ is the point $-P = (x_P, x_P + y_P)$. If P and Q are distinct points such that P is not $-Q$, then $P + Q = R$ where

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

$$x_R = (\lambda^2 + \lambda + x_P + x_Q + a),$$

$y_R = \lambda(x_P + x_R) + x_R + y_P,$

As with EC groups over real numbers, $P + (-P) = O$, the point at infinity. Furthermore, $P + O = P$ for all points P in the EC group.

Definition (Doubling the point in $F_{٢}^{m}$) (٢.٣٧)

If $x_P =$ ٠, then $٢P = O$, when x_P is not ٠, $٢P = R$ where

$$\lambda = x_P + \frac{y_P}{x_P},$$

$$x_R = \lambda^2 + \lambda + a,$$

$$y_R = x_P + (\lambda + 1)x_R,$$

Recall that a is one of the parameters chosen with the EC and that λ is the slope of the line through P and Q.

Example (٢.١٠)

An EC over $F_{٢}^{٤}$. A polynomial basis representation is used for the elements of $F_{٢}^{٤}$

Consider the field $F_{٢}^{٤}$ generated by the root $g = x$ of the irreducible polynomial $f(x) = x^{٤} + x + ١$.

Consider the non-supersingular EC over $F_{٢}^{٤}$ with defining equation $y^{٢} + xy = x^{٣} + g^{٤}x^{٢} + ١$

Then the solutions over $F_{٢}^{٤}$ to the equation of the EC are: $(٠, ١), (١, g^{٦}), (١, g^{١٣}), (g^{٣}, g^{٨}), (g^{٣}, g^{١٣}), (g^{٥}, g^{٣}), (g^{٥}, g^{١١}), (g^{٦}, g^{٨}), (g^{٦}, g^{١٤}), (g^{٩}, g^{١٠}), (g^{٩}, g^{١٣}), (g^{١٠}, g^{١}), (g^{١٠}, g^{٨}), (g^{١٢}, ٠), (g^{١٢}, g^{١٢})$.

The group $E(F_{٢}^{٤})$ has ١٦ points (including the point at infinity O). The following are examples of the group operation.

(١) Let $P_{١} = (x_{١}, y_{١}) = (g^{٦}, g^{٨}),$

$$P_2 = (x_2, y_2) = (g^3, g^{13}),$$

$$P_1 + P_2 = (x_3, y_3),$$

then $\lambda = \dfrac{y_1 - y_2}{x_1 - x_2} = g,$

$$x_{٣} = \lambda^{٢} + \lambda + x_{١} + x_{٢} + a = g^{٢} + g + g^{٦} + g^{٣} + g^{٤} = ١,$$

$$y_{٣} = \lambda(x_{١} + x_{٣}) + x_{٣} + y_{١} = g\,(g^{٦} + ١) + ١ + g^{٨} = g^{١٣}.$$

(٢) If $٢P_{١} = (x_{٣}, y_{٣})$, then

$$\lambda = x_1 + \frac{y_1}{x_1} = g^6 + \frac{g^8}{g^6} = g^3 ,$$

$$x_3 = \lambda^2 + \lambda + a = g^6 + g^3 + g^4 = g^{10} ,$$

$$y_3 = x_1^2 + (\lambda + 1)x_3 = g^{12} + (g^3 + 1)g^{10} = g^8 .$$

(٢. ٥. ٢) Fast Group Operations on Elliptic Curves

The most fundamental computation on ECs is the group operations of the type[٤٦] $kP = \underbrace{P \oplus P \oplus ... \oplus P}_{k \quad times}$.

where $P = (x, y)$ is a point on an EC E, and k a very large positive integer. Since the computation of kP is so fundamental in all EC related computations and application, it is desirable that such computations are carried out as fast as possible. The idea of ***repeated squaring*** for fast exponentiations can be used almost directly for fast group operations (i.e., fast point additions) on ECs. The idea of fast group additions is as follows: [٤٢]

– Translate k to binary form, so let $k = e_{٠} e_{١} e_{٢} \ldots e_r$.

– then for i starting from $e_{٠}$ down to $e_{r-١}$ (e_r is always ١ and use for initialization [٤٦]) , if e_i = ١ ,then perform a doubling and an addition group operation; otherwise , just perform a doubling operation.

Example (٢. ١١)

Compute ٨٩ P.

Solution

– the binary representation of ٨٩ is ١٠١١٠٠١, so we have

e_6	1	P	*initialization*
e_5	0	$2P$	*doubling*
e_4	1	$2(2P)+P$	*doubling and addition*
e_3	1	$2(2(2P)+P)+P$	*doubling and addition*
e_2	0	$2(2(2(2P)+P)+P)$	*doubling*
e_1	0	$2(2(2(2(2P)+P)+P))$	*doubling*
e_0	1	$\underbrace{2(2(2(2(2(2P)+P)+P)))+P}_{\Downarrow \atop 89P}$	*doubling and addtion*

On top of that a doubling operation is slightly faster than a summing operation. Koblitz calls this a " ***balanced*** " expansion. The algorithm converts a string of set bits to a string of zero bits followed by –١. to make this clear, we will take another example:

١٠٠٤٥ P = (١٠٠١١١٠٠١١١١٠١) P.

The last bit in the chain of set bits is replaced with –١ , all the other bits are replaced with ٠'s**,** and the leading ٠ is set. So the balanced representation becomes : ١ ٠ ١ ٠ ٠ –١ ٠ ١ ٠ ٠ ٠ –١ ٠ ١ and the operation are :

$(((((2P * 2 + P)2 * 2 * 2 - P)2 * 2 + P)2 * 2 * 2 * 2 - P)2 * 2 + P).$

(٢. ٥. ٣) Number of Points on Elliptic Curves

In this section , we shall study the problem which say :How many points are there on an EC ?

Theorem (٢. ٤): [٣٧], [٣]

There are $1+\sum_{x\in F_p}\left(1+\left(\frac{x^3+ax+b}{p}\right)\right)$ points on $E: y^{٢} = x^{٣}+ax+b$, including the point at infinity O, where $\left(\frac{x^3+ax+b}{p}\right)$ is the Legendre symbol.

Example (٢. ١٢)

let p = ٥**,** all the possibilities occur in the following ECs given in the following table:

Elliptic Curves	Number of points	Elliptic Curves	Number of points
$y^{٢} = x^{٣} + ١$	٦	$y^{٢} = x^{٣} + ٢x + ١$	٧
$y^{٢} = x^{٣} + ٢$	٦	$y^{٢} = x^{٣} + ٢x + ٤$	٧
$y^{٢} = x^{٣} + ٣$	٦	$y^{٢} = x^{٣} + ٣x$	١٠
$y^{٢} = x^{٣} + ٤$	٦	$y^{٢} = x^{٣} + ٣x + ٢$	٥
$y^{٢} = x^{٣} + x$	٤	$y^{٢} = x^{٣} + ٣x + ٣$	٥
$y^{٢} = x^{٣} + x + ١$	٩	$y^{٢} = x^{٣} + ٤x$	٨
$y^{٢} = x^{٣} + x + ٢$	٤	$y^{٢} = x^{٣} + ٤x + ١$	٨
$y^{٢} = x^{٣} + x + ٣$	٤	$y^{٢} = x^{٣} + ٤x + ٢$	٣

$y^{٢} = x^{٣} + x + ٤$	٩	$y^{٢} = x^{٣} + ٤x + ٣$	٣
$y^{٢} = x^{٣} + ٢x$	٢	$y^{٢} = x^{٣} + ٤x + ٤$	٨

Theorem (٢.٥)[٤٣]

Let $E(F_p)$ is an EC E defined over the finite field F_q have prime order $\#E$, then for all $P \in E(F_q)$ and $P \neq O$, then P have the order $m = \#E$. Then P generates subgroup equal to $E(F_q)$.

(٢.٦) Elliptic Curve Discrete Logarithm Problem

The security of ECCs relies on the difficulty of solving the *ECDLP* [٥]. The motivation of using ECs in cryptography is that there is no known sub-exponential algorithm, which solves the ***ECDLP*** in general [٥٥].

Let us state it formally as the following:

"Given an EC E defined over a field F, a point $P \in E(F)$ of order n and a point $Q \in E(F)$, determine the integer l, $٠ \leq l \leq n - ١$, such that $Q = kP$, provided that such an integer exists ".

The level of difficulty of solving the ECDLP may be lowered. These are precisely the situations that the ECCs must avoid in order that its security level is ensured.

Example(٢.١٣)

In the EC group defined by $y^{٢} = x^{٣} + ٩x + ١٧$ over $F_{٢٣}$, we will compute the discrete logarithm k of $Q = (٤, ٥)$ to the base $P = (١٦, ٥)$.

Solution

One way to find k is to compute multiples of P until Q is found. The first few multiples of P are:

$P = (١٦,٥)$, $٢P = (٢٠,٢٠)$, $٣P = (١٤,١٤)$, $٤P = (١٩,٢٠)$,
$٥P = (١٣,١٠)$, $٦P = (٧,٣)$, $٧P = (٨,٧)$, $٨P = (١٢,١٧)$, $٩P = (٤,٥)$

Since $٩P = (٤,٥) = Q$,

the discrete logarithm of Q to the base P is $k = ٩$.

In a real application, ***k*** would be large enough such that it would be infeasible to determine ***k*** in this manner.

Chapter Three
Elliptic Curve Cryptosystem

(٣. ١) Introduction

ECCs have the potential to provide relatively small block size, high-security public key schemes that can be efficiently implemented [٣٢]. The core of the ECCs is when it is used with Galois Field it becomes a one – way function [١] [١٩].

In this chapter, we describe the two different classes of cryptosystem, namely, private and public key cryptosystem, following the detailed of the ECCs and the attacks of it.

To start with, the chapter we must define some commonly used terms that are required for our discussions.

Definition (٣. ١)[٣]

A ***plaintext*** is a user-readable meaningful message or data, while a ***ciphertext*** is an unintelligible modified from the plaintext.

Definition (٣. ٢)[١]

The process of converting from plaintext to ciphertext is said to be ***encryption*** (***enciphering***), an unintelligible modified from the plaintext, while the reverse process of changing from ciphertext to plaintext is called ***decryption*** (***deciphering***).

From this point on, let $\mathcal{M}$ denoted the set of all possible plaintext, $\mathcal{C}$ the set of all possible ciphertext and $\mathcal{K}$ the set of all possible keys [٣]. In addition, we shall adopt two people, Ali and Benin, in our illustrations.

[١] which we remember it in section (٣.٣) .

(٣. ٢) Private Key Cryptosystem

Definition (٣. ٣)[٣٠]

A ***private key cryptosystem***, consists of a family of pairs of functions $E_k : \mathcal{M} \rightarrow \mathcal{C}$, $D_k : \mathcal{C} \rightarrow \mathcal{M}$, $k \in \mathcal{K}$, such that $D_k(E_k(m)) = m$ for all $m \in \mathcal{M}$ and $k \in \mathcal{K}$.

To illustration this definition, let us take a particular private key cryptosystem as an example.

Suppose that [٢] Ali and Benin agree initially upon secret key $k \in \mathcal{K}$. if Ali wants to send a message $m \in \mathcal{M}$, he sends the ciphertext $c = E_k(m)$ to Benin . Benin then recovers m by applying the decryption function D_k . it should be infeasible for an eavesdropper who sees c to determine the message m or the key k.

In secret key (private key) cryptography, the sender and the receiver use the same secret key. The main problem is to exchange this secret key. You can think for example at what happens if a large amount of users have to communicate using a secret key cryptosystem [٥٦].

(٣. ٣) Public Key Cryptosystem

Definition (٣. ٤)[٥٧]

A ***one – way function*** is a function f such that for each x in the domain of f , it is easy to compute $f(x)$; but for essentially all y in the range of f , it is computationally infeasible to find any x such that $y = f(x)$.

Definition (٣. ٥)[٣],[٣٠]

A one-way function $f : \mathcal{M} \rightarrow \mathcal{C}$ is said to be a ***trapdoor one – way function*** (TOF) if there exist some information with which f can be efficiently inverted. This extra information is called the ***trapdoor***.

Example (٣. ١)

The function $f_{k,N} : x \mapsto x^k \bmod N$ is ***TOF***, where $N = pq$ with p and q primes, and $kk' \equiv 1 \pmod{\Phi(N)}$. it is obvious that f is easy

[٢] The reader can take more information on this example in [٤٦].

to compute since the modular exponentiation x^k ***mod N*** is easy to compute . but f^{-1}, the inverse of ***f*** (i.e the ***kth*** root of x modulo N) is difficult to compute . However, if k', the trapdoor is given, ***f*** can be easily inverted, since $(x^k)^{k'} = x$.

Definition (٣. ٦)[٣]

A ***public key cryptosystem***, consists of a family $f_k : \mathcal{M} \to \mathcal{C}$, $k \in K$, of trapdoor one-way functions.

In public key cryptography, each user X has a private key d_X (nobody except X knows this key), and a public key e_X which is published and can be known by everybody [٥٦] . ($d_X , e_X \in \mathcal{K}$ such that $d_X \circ e_X$ = ***identity***)

When Ali wants to send a message $M \in \mathcal{M}$ to Benin, Ali uses the public key of Benin to compute the ciphertext $C \in \mathcal{C}$, as follows :

$$C = e_{Benin}(M).$$

The ciphertext C is sent to Benin who is able to recover the message (plaintext) as follows :

$$\begin{aligned} M &= d_{Benin}(C) \\ &= d_{Benin}(e_{Benin}(M)) \\ &= M. \end{aligned}$$

Public key cryptosystem have some important advantages over private key cryptosystem in the distribution of the keys. However, when a large amount of information has to be communicated, it may be that the use of public key cryptography would be too slow whereas the use of secret key cryptography could be impossible for the lack of a shared secret key [٤٦] .

(٣. ٤) Elliptic Curve Cryptography

In ١٩٧٧, R. Rivest, A. Shamir, and L. Adleman proposed a public key cryptosystem, which uses only elementary ideas from number theory. Their enciphering system is called RSA, [١] .

While ECC first proposed in ١٩٨٥ independently by Neal Koblitz from the University of Washington, and Victor Miller, who was then at IBM, Yorktown Heights [٤],[١٦]. They did not invent a new cryptographic algorithm with ECs over finite fields, but they implemented existing algorithms, like Diffie – Hellman, using ECs [٣٠]. This cryptosystem based on DLP in the group of points of an

EC defined over a finite field. Hence, ECC can match the security of the other cryptosystems while using smaller key [٩].

In this subsection we present the important terms which is the embedding any number as a point in the EC, and generating a point of an EC, and follows the all ECCss and attacks of it, with examples for illustrating.

(٣. ٤. ١) Embedding Plaintext on an Elliptic Curve

Suppose we would like to encrypt some plaintext with ***ECC***. There has to be a method, which takes some arbitrary text and embedded it in an EC, i.e. which gives a bijection between the points on an EC and a plaintext block [١٣].

Now we can embed any message as a point in an EC defined over a finite field F_p, with p is prime as follows [٤٦]:–

Let our message units m be integers $٠ \leq m \leq M$ where M is the size of the message block, let also k be a large enough integer for us to satisfy with an error probability of ٢^{-k} when we attempt to embed a plaintext message m. In practice, $٣٠ \leq k \leq ٥٠$. Now let us take $k =$ ٣٠ and an EC $E : y^٢ = x^٣ + ax + b$ over F_p.

Given a message number m, we compute a set of value for x as follows:–

$$X = \{ mk + j ,\ \ j = ٠,\ ١,\ ٢, ...\}$$
$$= \{ ٣٠m ,\ ٣٠m + ١,\ ٣٠m + ٢, ...\}$$ until we find $x^٣ + ax + b$ is a square modulo p , giving us point $(x ,\ \sqrt{x^3 + ax + b}\ )$ on E.

To convert a point (x, y) on E back to a message number m, we just compute $m = \lfloor x/30 \rfloor$.

Example (٣. ١)

Let E: $y^٢ = x^٣ + ٨٣٨٠\, x + ٨٣٨٦$, be an EC defined over $F_{٨٣٨٧}$. Embed the message $m =$ ١٢٣ as a point on E .

Solution

Suppose that $k =$ ٣٠ , then we can compute the point (x, y) as follows:–

$x = \{ 30 \cdot 123 + j ,\quad j = 0 ,1,2 ,3 ,... \}$, until $x^٣ + ٨٣٨٠\, x + ٨٣٨٦$ is square modulo ٨٣٨٧. We find that when $j =$ ٣:

$$x = (٣٠ \cdot ١٢٣ + ٣) mod\ ٨٣٨٧$$
$$= ٣٦٩٣$$
$$x^٣ + ٨٣٨٠ \cdot x + ٨٣٨٦ = (٣٦٩٣)^٣ + ٨٣٨٠ \cdot ٣٦٩٣ + ٨٣٨٦$$
$$\equiv ٣٠٩٦٠٧٥٩\ mod\ ٨٣٨٧$$
$$= ٣١٨٥^٢\ mod\ ٨٣٨٧$$

So we get the message point for m = ١٢٣:

$$(x, \sqrt{x^3 + ax + b}) = (3693, 3185)$$

To convert the message point $(3693, 3185)$ on E back to its original message number m, we just compute

$$m = \lfloor 3693/30 \rfloor = \lfloor 123.1000 \rfloor = 123.$$

(٣. ٤. ٢) Generating a Point of an Elliptic Curve

One open question is how to find a point on an EC .Suppose we would like to find an arbitrary point. Then conceder to reference [١٣], we can do the following:

- Choose x
- Compute $z = y^{٢} = x^{٣} + a\,x + b$
- If z = ٠, then y = ٠ and there is only one point, $(x,\ ٠)$ with the given $x-$coordinate.

Else

- Verify whether there exists y such that $z \equiv y^{٢}\ (mod\ p)$ (i.e. check whether z is quadratic residue $mod\ p$), then by ***theorem (٢. ٢)***, we can check whether $z^{(p-١)/٢} \equiv ١\ (mod\ p)$. Now if this condition is satisfying, then there are two points with a given $x-$coordinate $(x, y_{١})$ and $(x, y_{٢})$.
  Where

$$y_{١} = z^{(p+١)/٤}\ (mod\ p)$$
$$y_{٢} = -z^{(p+١)/٤}\ (mod\ p)$$
$$\equiv (p - z^{(p+١)/٤})\ (mod\ p)$$
$$= p - y_{١}.$$

Else

- If $z^{(p-١)/٢} \cong ١\ (mod\ p)$, then there is no point of the given elliptic curve with the given $x-$coordinate.

Example (٣. ٢)

Let $E\!:\ y^{٢} = x^{٣} + x + ١$ be an EC defined over $F_{٢٣}$. In addition, suppose we would like to find a point on E with $x-$coordinate is ٣.

Solution:

$$z = (٣^{٣} + ١ \times ٣ + ١) mod\ ٢٣ = ٨$$

By ***theorem (٢. ٢)***

$$٨^{(٢٣-١)/٢} (mod\ ٢٣) = ٨^{١١}\ mod\ ٢٣$$
$$= (٨^{٨}\ mod\ ٢٣)(٨^{٢} mod\ ٢٣)(٨^{١}\ mod\ ٢٣)(mod\ ٢٣)$$
$$= ٤ \times ١٨ \times ٨\ (mod\ ٢٣)$$
$$= ١$$

Then there are two points with a given $x-$ coordinate $(x, y_{١})$ and $(x, y_{٢})$.

Such that $y_{١} = ٨^{(٢٣+١/٤)} \bmod ٢٣$

$= ٨^{٦} \bmod ٢٣$

$= ١٣$

and $y_{٢} = -١٣$

$\equiv (٢٣-١٣) \bmod ٢٣$

$= ١٠.$

Then the two points are $(٣, ١٣), (٣, ١٠)$.

(٣.٥) Cryptosystem Based on Elliptic Curve Philosophy

In what follows, we shall introduce the three public key cryptosystems widely used ECs, namely the Diffie – Hellman Key Exchange system, the Massey – Omura, and the ElGamal public key cryptosystems.

(٣.٥.١) Elliptic Curve Diffie–Hellman (ECDH) Key Exchange

The Diffie-Hellman key agreement protocol was developed by Diffie and Hellman in ١٩٧٦. and published in the groundbreaking paper “***New Directions in Cryptography***” .In that paper, they also introduced the revolutionary concept of public key cryptography [٣]. This system widely used as ECCs.

This system is merely a method for exchanging key; no messages are involved. The following algorithm illustrates this system .

Algorithm for ECDH

Setup

- Ali and Benin agree upon an EC $E(F_p)$ and a base point B.
- Ali (Benin) chooses a random and secret $e(d)$ (such that $١ \leq e, d \leq \#E)$, and computes $eB(dB)$ and sends it to Benin(Ali).

  Then eB and dB are public and e and d are a secret.
- Ali (Benin) computes the secret key edB (deB).

Without solving the ECDLP, there is no fast way to compute edB if only knows B, eB and dB [٤٦]. After these setups, Ali and Benin have the same point (only Ali and Benin know it).

Example (٣.٣):

Let ***E*** be an EC define over ***F***$_{٨٢٢٣}$ with parameters ***a*** = ٠, ***b*** = ١٣٩ where ***(***٤***a***^{٣}+٢٧***b***^{٢}***)mod p*** = ٢٩٨٨٨≠ ٠. and ***# E*** = ٨٠٨٩.

Since ***#E*** is prime number then by ***theorem (***٢.٥***),*** every point on ***E*** is base point, then let ***B*** = ***(***٨٢١٦ , ٧٤٧٧***).*** Suppose Ali and Benin agree upon an EC ***E(F***$_{٨٢٢٣}$***)*** and a base point ***B***. how they can exchanging any key?

Solution

- Ali chooses a secret random integer ***e*** = ٦٢٣٤.

  eB = ٦٢٣٤ * ***(***٨٢١٦,٧٤٧٧***)***

  = ***(***٤١٠٧, ٦٣٦٤***),*** and send ***(***٤١٠٧, ٦٣٦٤) to Benin .

Benin chooses a secret random integer ***d*** = ٣٥٤١.

dB = ٣٥٤١ * ***(***٨٢١٦,٧٤٧٧***)***

= ***(***١١٧٧, ٦٢٨٤***),*** and send ***(***١١٧٧, ٦٢٨٤***)*** to Ali

- Ali computes the secret key ***e (dB)*** :

  e(dB) = ٦٢٣٤ * ***(***١١٧٧, ٦٢٨٤***)***

  = ***(***٥٣٤١, ٥٢٧٥***)***

Benin computes the secret key ***d (eB)*** :

d(eB) = ٣٥٤١***(***٤١٠٧, ٦٣٦٤***)***

= ***(***٥٣٤١, ٥٢٧٥***)***

Now, Ali and Benin have the same point ***edB*** = ***(***٥٣٤١, ٥٢٧٥***).***

(٣.٥.٢) Elliptic Curve Massey –Omura (ECMO) Cryptosystem

In this system, the finite field ***F***$_{p}$ and the EC ***E*** have been made publicly known. The number ***#E*** has been compute and is also publicly known [٤٣].

The ECMO like the ECDH depends on the difficulty of solving the ECDLP*,* and like the most of the public key cryptosystems. However, it is different from ECDH*,* since the ECDH using for exchanging key; no messages are involved, while the ECMO using to send a message which must be a point on an EC, and there is not base point . The following algorithm illustrates this system [٤٢].

Algorithm for ECMO

If Ali wants to send Benin the message ***M***, then they do the following setup:

Setup

- Ali and Benin agree upon an EC ***E(F***$_{p}$***)*** with ***#E*** = ***N***.

- Ali chooses a random and secret $e_{Ali} \in [١, N]$ such that $gcd\ (e_{Ali}\ , N) = ١$, then he computes $(e_{Ali}\ M)$ and sends it to Benin.
- Benin chooses a random and secret $e_{Benin} \in [١, N]$ such that $gcd\ (e_{Benin}, N) = ١$, then she computes $(e_{Benin}\ (e_{Ali}\ M))$ and sends it to Ali.
- Ali computes $d_{Ali}=e^{-1}_{Ali}(mod\ N)$, then computes $d_{Ali}(e_{Benin}e_{Ali}\ M) = (e_{Benin}\ M)$ and sends it to Benin.
- Benin computes $d_{Benin} = e^{-1}_{Benin}(mod\ N)$, then she converts the message M by computing the follwing:

$$\begin{aligned} d_{Benin}\ d_{Ali}\ (e_{Benin}\ e_{Ali}\ M) &= d_{Benin}\ e_{Benin}\ M \\ &= M. \end{aligned}$$

Note that an eavesdropper would know $e_{Ali}\ M,\ e_{Benin}\ (e_{Ali}\ M)$, and $(e_{Benin}\ M)$. So if he could solve the ECDLP, he could determine e_{Benin} from the first two points and compute $d_{Benin} = e^{-1}_{Benin}(mod\ N)$ and hence get $M = d_{Benin}\ (e_{Benin}\ M)$ [٤٦]. Nevertheless, as we say the ECDLP is intractable.

Example (٣.٤)

Let E be an EC define over $F_{١٠٦١٣}$ with parameters $a = ٠$, $b = ٨٦$ where $(٤a^{٣}+ ٢٧b^{٢})mod\ p = ٨٦٥٨ \neq ٠$. and $\# E = ١٠٦١٤$.
If Ali wishes to send the message $M = (٤٩٤٩,\ ٤٧١٥)$ to Benin using ***ECMO*** cryptosystem, what should they do?

Solution

- Ali chooses a random and secret $e_{Ali} \in [١, N]$ such that $gcd\ (e_{Ali}\ , N) = ١$, let $e_{Ali} = ٨٤٢٥$. Then he computes
  $$\begin{aligned} e_{Ali}\ M &= ٨٤٢٥ * (٤٩٤٩,\ ٤٧١٥) \\ &= (٥١١٩,\ ٥٨٢٧) \end{aligned}$$
  and send it to Benin.
- Benin chooses a random and secret $e_{Benin} \in [١, N]$ such that $gcd\ (e_{Benin}, N) = ١$, ***let*** $e_{Benin} = ٧٥٢٣.$
  Then she computes $e_{Benin}\ (e_{Ali}\ M) = ٧٥٢٣ * (٥١١٩,\ ٥٨٢٧)$
  $= (٤٣٨٢,\ ٧٥٤)$,
  and sends it to Ali.
- Ali computes $d_{Ali} = e^{-1}_{Ali}(mod\ N)$
  $= ٨٢٠٩,$ Then computes
  $$\begin{aligned} d_{Ali}\ (e_{Benin}\ e_{Ali}\ M) &= ٨٢٠٩ * (٤٣٨٢,\ ٧٥٤) \\ &= (٧٢٢,\ ٣٣٣٠) \end{aligned}$$
  and send it to Benin.
- Benin computes $d_{Benin} = e^{-1}_{Benin}(mod\ N),$
  $= ١٥٨٣,$

Then she converts the message ***M*** by computing

$d_{Benin}\, d_{Ali}\, (e_{Benin}\, e_{Ali}\, M)$ = ١٥٨٣* (٧٢٢, ٣٣٣٠)
= (٤٩٤٩, ٤٧١٥)
= ***M***.

(٣. ٥. ٣) ElGamal Elliptic Curve Cryptosystem

This system is very simple for two commutations (sender and receiver) in cryptography operation, since there is one sender operation and one receiver operation. Now let us take the following algorithm to illustrate this system.

Algorithm for ElGamal Elliptic Curve Cryptosystem

If Ali wants to encrypt and send Benin the message ***M***, then they do the following setup:

Setup

- Ali and Benin agree upon an EC $E(F_p)$ and a base point ***B***.
- Benin chooses a random integer ***d*** as a secret key, and computes her public key ***dB*** .
- Ali chooses a random integer ***e*** as a secret key, and produce the ciphertext C_m consisting of the pair of points ($C_m = \{ C, eB\}$) and send it to Benin , where $C = M + e\,(dB)$.
- If Benin wishes to decrypt the ciphertext, she multiplies the second point of the pair by her secret key and subtracts the result from the first point:

$$C - d\,(e\,B) = M + e(dB) - d\,(e\,B)$$
$$= M.$$

An eavesdropper who can solve the ECDLP can of course, determine ***d*** from the publicly known information ***B*** and ***dB***. But as everybody knows, there is no efficient way to compute discrete logarithms, so the system is secure [٤٦].

Example (٣. ٥)

Let ***E*** be an EC define over ***F***₍١١٣١٧₎ with parameters ***a*** = ٩٨١٧, ***b*** = ٤٧ where (٤*a*^٣+ ٢٧ *b*^٢) ***mod p*** = ٧٠٩٠ ≠ ٠. And ***# E*** = ١١٤٨٩ . Since ***#E*** is prime number then by ***theorem (٢. ٥)***, every point on ***E*** is base point, then let ***B*** = ***(١١١١٧, ٣٦٦٣).***

If Ali wishes to send the message ***M*** = ***(١٠٤٩٨, ١٣٠٤)*** to Benin using ElGamal ECCs, what should they do?

Solution

- Benin chooses a random integer ***d*** as a secret key, let ***d*** = ٧٣٩١, and computes her public key

dB = ٧٣٩١ * (١١١١٧, ٣٦٦٣)
= (٨٩١٦, ٧٥٥٢).

- Ali chooses a random integer ***e*** as a secret key, let ***e*** = ٦٦٩٣ and computes ***e(dB)*** = ٦٦٩٣ * (٨٩١٦, ٧٥٥٢)
= (٢٠٩٤, ٦١٤٥),
and computes ***eB*** = ٦٦٩٣ * (١١١١٧, ٣٦٦٣)
= (٣٢٦, ٢٤١٧),
and computes ***C*** = ***M*** + ***e (dB)***
= (١٠٤٩٨, ١٣٠٤) + (٢٠٩٤, ٦١٤٥)
= (٣٠٣٨, ٣٦٧),
and sends {(٣٠٣٨, ٣٦٧), (٣٢٦, ٢٤١٧)} to Benin .
- Benin to decrypt the ciphertext,
she computes ***d (e B)*** = ٧٣٩١ * (٣٢٦, ٢٤١٧)
= (٢٠٩٤, ٦١٤٥),
and computes ***C*** – ***d (e B)*** = (٣٠٣٨, ٣٦٧) – (٢٠٩٤, ٦١٤٥)
= (٣٠٣٨, ٣٦٧) + (٢٠٩٤, – ٦١٤٥)
= (١٠٤٩٨, ١٣٠٤)
= ***M***.

The above are some of ECCss of certain public key cryptosystems. It should be noted that almost every public key cryptosystem has an EC analogue; it is of course possible to develop new ECCss that do not rely on the existing cryptosystems.

(٣.٦) Menezes–Vanstone Elliptic Curve Cryptosystem(MVECC)

This is cryptosystem that has no analogue for DLP [٤٣] (i.e. this cryptosystem dose not depend on DLP as the above cryptosystems).

In this variation (***MVECC***), the EC is used for “masking”, and plaintexts and ciphertexts are allowed to be arbitrary ordered pairs of (nonzero) elements (i.e., they are not required to be points on ***E***) [٥٨] , [٣٠], [٧]. Now let us take the following algorithm to illustrate this system.

Algorithm for MVECC

If Ali wants to encrypt and send Benin the message ***M***, then they do the following setup

Setup:

- Ali and Benin agree upon an EC ***E(F***$_p$***)*** and a base point ***B***.

- Benin first selects a private key d and generates a public key $Q = d\,B.$
- Ali wishes to encrypt and send a message $M=(m_{١},m_{٢}).$ to Benin, he chooses a random positive integer e and produces the ciphertext C_m consisting of the pair of points ($C_m = \{ C, eB\}$) and send it to Benin , where $C = (c_{١}, c_{٢})$ and where

$$c_{١} = m_{١} * k_{١} \bmod p,$$
$$c_{٢} = m_{٢} * k_{٢} \bmod p.$$
$$eQ = (k_{١},k_{٢})$$

- Benin likes to decrypt the ciphertext, she computes the following:

$$(k_{١},k_{٢}) = d\,(eB), \text{ and then}$$
$$m_{١} = c_{١} * k_{١}^{-١} \bmod p$$
$$m_{٢} = c_{٢} * k_{٢}^{-١} \bmod p.$$

Example (٣. ٦)

Let E be an EC define over $F_{١٨٤٦١}$ with parameters $a =$ ١٧٩٦١, $b =$ ١٧٩٦٧ where $(٤a^{٣}+ ٢٧b^{٢}) \bmod p =$ ١٤٥٨٠ $\neq$ ٠. And $\# E =$ ١٨٦٧١. If Ali wants to send the message $M =$ (٥٦٣١, ٨٤١٩) to Benin using ***MVELC*** cryptosystem, what should they do?

Solution

Since $\#E$ is prime number then by ***theorem (٢. ٥)***, every point on E is base point, then let $B =$ (١٨٢٨٧, ١٧٩٦٥).

- Benin first selects a private key d, let $d =$ ٨٩٥ and generates a public key $Q = d\,B$

$$= ٨٩٥*(١٨٢٨٧, ١٧٩٦٥)$$
$$= (١٥٣٢٨, ١٦٣٠) .$$

- If Ali wishes to encrypt and send a message $M=$(٥٦٣١, ٨٤١٩) $= (m_{١}, m_{٢}).$ to Benin, he chooses a random positive integer e , let $e =$ ٧٢٣ and produces the ciphertext C_m

He computes $eB =$ ٧٢٣ * (١٨٢٨٧, ١٧٩٦٥)

$$= (١٣٣٤٠, ١٦٢٥٨)$$

He computes $eQ =$ ٧٢٣ * (١٥٣٢٨, ١٦٣٠)

$$= (١٠٧١٣, ٨٠٤٢)$$
$$= (k_{١},k_{٢}),$$

and then he computes $c_{١} = m_{١} * k_{١} \bmod p$

$$= ٥٦٣١ * ١٠٧١٣ \bmod ١٨٤٦١$$
$$= ١٢٨١٦$$
$$c_{٢} = m_{٢} * k_{٢} \bmod p$$
$$= ٨٤١٩ * ٨٠٤٢ \bmod ١٨٤٦١$$
$$= ٩١١١,$$

and send $C_m = \{(١٢٨١٦, ٩١١١), (١٣٣٤٠, ١٦٢٥٨)\}$.

- If Benin would like to decrypt the ciphertext, she computes the following:

$$
\begin{aligned}
(k_{١},k_{٢}) &= d\,(eB)\\
&= ٨٩٥\,(١٣٣٤٠,\ ١٦٢٥٨)\\
&= (١٠٧١٣,\ ٨٠٤٢),\\
k_1^{-1} &= ١٠٧١٣^{-١} \bmod ١٨٤٦١\\
&= ١٥٢٤٢,\\
k_1^{-1} &= ٨٠٤٢^{-١} \bmod ١٨٤٦١\\
&= ٨٣٤٩,
\end{aligned}
$$

then $m_{١} = c_{١} * k_{١}^{-١} \bmod p$

$$
\begin{aligned}
&= ١٢٨١٦ * ١٥٢٤٢ \bmod ١٨٤٦١\\
&= ٥٦٣١\\
m_{٢} &= c_{٢} * k_{٢}^{-١} \bmod p\\
&= ٩١١١ * ٨٣٤٩ \bmod ١٨٤٦١\\
&= ٨٤١٩
\end{aligned}
$$

(٣.٧) Elliptic Curve Digital Signature Algorithm

The ***DSA*** was proposed in August ١٩٩١ by the U.S. National Institute of Standards and Technology (***NIST***) [٥٩] . The ***ECDSA*** is the EC analog of the ***DSA*** [٢٤]. ***ECDSA*** was first proposed in ١٩٩٢ by Vanstone [١٣].

A digital signature is a number that depends on the secret key only known by the signer and on the contents of the message being signed.

We briefly outline the ***ECDSA*** below [٦٠],[٥٩].

First, suppose that E be an EC defined over F , and a base point P of order n are selected and made public to all users (Ali and Benin).

ECDSA Key Generation The user Ali follows these steps:

- Select a random integer $d \in [٢, n-٢]$.
- Compute $Q = d * P$.
- The public and private keys of the user Ali are (E, P, n,Q) and d, respectively.

ECDSA Signature Generation The user Ali signs the message M using these steps:

- Select a random integer $k \in [٢, n-٢]$.   **
- Compute $k * P = (x_{١}, y_{١})$ and $r = x_{١} \bmod n$.

  If $r = ٠$ then go to Step **.

- Compute $k^{-١}$ ***mod n.***
- Compute $s = k^{-١}(H(M) + dr) \bmod n.$

  Here H is the secure hash algorithm ***SHA***.

  If $s = ٠$ go to Step **.
- The signature for the message M is the pair of integers ***(r, s).***

ECDSA Signature Verification The user Benin verifies Ali's signature (r, s) on the message M by applying the following steps:

- Compute $c = s^{-١} \bmod n$ and $H(M).$
- Compute $u_١ = H(M)c \bmod n$ and $u_٢ = rc \bmod n.$
- Compute $u_١ * P + u_٢ * Q = (x_٠, y_٠)$ and $v = x_٠ \bmod n.$
- Accept the signature if $v = r.$

If the signature (r, s) on the message M was indeed generated by Ali , the $s = k^{-١}(H(M)+dr) \bmod n$. With this information we have

$$\begin{aligned} k &\equiv s^{-١}(H(M) + dr) \\ &\equiv (s^{-١} H(M) + s^{-١}rd)\bmod n \\ &\equiv (u_١ + u_٢d) \bmod n \end{aligned}$$

Thus $u_١P + u_٢Q = (u_١ + u_٢d)P$

$= Kp,$ *and* so $v = r$ as required [١٣].

Example (٣. ٦)

Suppose that Ali and Benin agree upon an EC $E: y^٢ = x^٣ + ١١٦٤٧x + ١١٦٦١$ over $F_{١٣١٤٧}$ and a base point $P = (١١٦٥١, ١٢٠٢٨)$, with order $n = ١٣١٦٣$. If Ali wants to send a digitally signed message to Benin, what did they do?

Solution

ECDSA Key Generation The user Ali follows these steps:

- Select a random integer $d \in [٢, n-٢]$. Let $d = ١٦٣٤$
- Compute $Q = d * P.$

  $= ١٦٣٤ * (١١٦٥١, ١٢٠٢٨)$

  $= (١٠٥٥٠, ٦٧٠٦)$
- The public and private keys of the user Ali are (E, P, n, Q) and d, respectively.

ECDSA Signature Generation The user Ali signs the message M using these steps:

- Select a random integer $k \in [٢, n-٢]$.let $k = ٤١٨٣$
- Compute $k * P = ٤١٨٣* (١١٦٥١, ١٢٠٢٨)$

  $= (٢٦٤٨, ١٠١٢٩)$

  $= (x_١, y_١)$

  And compute $r = x_١ \bmod n$

  $= ٢٦٤٨ \bmod ١٣١٦٣$

$= ٢٦٤٨$

Now $r \neq ٠$.

- Compute $k^{-١} \ mod \ n = ٤١٨٣^{-١} \ mod \ ١٣١٦٣$
  $= ٦٧٥٣.$
- let $H(M) = ٦١٢٤$,
  then $s = k^{-١}(H(M) + dr) \ mod \ n$
  $= (٦٧٥٣(٦١٢٤ + ١٦٣٤ * ٢٦٤٨)) \ mod \ ١٣١٦٣$
  $= ١١١١٥$
  now $s \neq ٠$.
- The signature for the message M is the pair of integers $(r, s) = (٢٦٤٨, ١١١١٥)$.

<u>ECDSA Signature Verification</u> The user Benin verifies Ali's signature (r, s) on the message M by applying the following steps:

- Compute $c = s^{-١} \ mod \ n$
  $= ١١١١٥^{-١} \ mod \ ١٣١٦٣$
  $= ٢٠٧٦$
- Compute $u_١ = H(M)c \ mod \ n$
  $= ٦١٢٤ * ٢٠٧٦ \ mod \ ١٣١٦٣$
  $= ١١١٢٩$
  and compute $u_٢ = rc \ mod \ n$
  $= ٢٦٤٨ * ٢٠٧٦ \ mod \ ١٣١٦٣$
  $= ٨٢٧٧$

Compute $u_١ * P + u_٢ * Q$

So, $u_١ * P + u_٢ * Q = ١١١٢٩*(١١٦٥١, ١٢٠٢٨)+٨٢٧٧ * (١٠٥٥٠, ٦٧٠٦)$
$= (١٥٥٢, ٥٨٠٩) + (٤٨٨, ٩٨٦٢)$
$= (٢٦٤٨, ١٠١٢٩)$
$= (x_٠, y_٠)$

and compute $v = x_٠ \ mod \ n = ٢٦٤٨ \ mod \ ١٣١٦٣$
$= ٢٦٤٨.$

- Now $v = r = ٢٦٤٨$, then Benin accepts this signature.

(٣.٨) Attacks on Elliptic Curve Cryptosystem

In this section we overview known attacks on the ECDLP and discuss how to avoid them in practice.

(٣.٨.١) Exhaustive Search

In exhaustive search, one simply computes successive multiples of ***P*: *P*, ٢*P*, ٣*P*, ٤*P*** until ***Q*** is obtained. This method can take up to ***n*** steps in the worst case. To circumvent this attack, one has to select EC parameters so that ***n*** is sufficiently large [٣٥] ,[٢٤].

Example (٣.٧)

What is the discrete logarithm of ***Q*** = ***(١٣٦٩٣, ٣٦٤٧)*** to the base ***P*** = ***(١٥٧٥٣, ١٤٣٣٦)*** in the EC $y^{٢} = x^{٣} + ١٤٨٢٣x + ١٥٧٨١$ over $F_{١٥٨٢٣}$

.

Solution

We have ***P*=(١٥٧٥٣, ١٤٣٣٦), ٢*P=(١٢٠٢٤, ٩٩١١), ٣*P=(٦٨١٢, ٧٤٢٥), ... , ٢٠ * P = (١٣٦٩٣, ٣٦٤٧)*** Then the discrete logarithm of ***Q*** to the base ***P*** is ٢٠ . This ***ECDLP*** is the only in this case, as we follows look that there is more of the ***ECDLP*** in the different cases, but there is no different since they have the same result.

(٣.٨.٢)Parallel Pollard rho Method

Pollards ρ method for computing discrete logarithms was first introduced in ١٩٧٨. This method makes use of the so- called "***birthday problem***"[٣] from statistics. [٦١]

The best attack known on the ECDLP is parallel collision search [٦٢], based on Pollard's ρ algorithm which has running time proportional to the square root of the largest prime factor dividing the curve order. This method works for any cyclic group and does not make use of any additional structure present in EC groups [٥]

(٣.٨.٢.١) Parallel Collision Search [٥]

Given a point ***Q*** on an EC which is in a subgroup of order ***n*** generated by ***P***, we seek ***l*** such that ***Q*** = ***lP***. Pollard's ρ method proceeds as follows. Partition the points on the curve into three roughly equal size sets ***S*₁ , *S*₂, *S*₃** based on some simple rule. Define an iteration function on a point ***Z*** as follows

$$f(Z)=\begin{cases} 2Z & if \quad Z \in S_1 \\ Z+P & if \quad Z \in S_2 \\ Z+Q & if \quad Z \in S_3. \end{cases}$$

Choose ***A*₀ ,*B*₀ ∈ [١, *n* – ١]** at random and compute the starting point ***Z*₀** = ***A*₀*P*** +***B*₀*Q***. Compute the sequence ***Z*₁** = ***f(Z*₀*)* , *Z*₂** = ***f(Z*₁*)* , . . .** keeping track of A_i, B_i

[٣] The birthday problem asks how many people need to be assembled together to have a ٥٠٪ chance that two of them share the same birthday. [٦١] such that $Z_i = A_i P + B_i Q$. Thus,

$$(Z_{i+1}, A_{i+1}, B_{i+1}) = \begin{cases} (2Z_i, 2A_i, 2B_i) & \text{if} \quad Z \in S_1 \\ (Z_i + P, A_i + 1, B_i) & \text{if} \quad Z \in S_2 \\ (Z_i + Q, A_i, B_i + 1) & \text{if} \quad Z \in S_3. \end{cases}$$

Note that A_i and B_i can be computed *modulo* n so that they do not grow out of control. Because the number of points on the curve is finite, the sequence of points must begin to repeat. Upon detection that $Z_i = Z_j$ we have $A_i P + B_i Q = A_j P + B_j Q,$

which gives $l = \dfrac{A_i - A_j}{B_j - B_i} \bmod n$ unless we are very unlucky and

$B_i \equiv B_j \ (\bmod n).$

Actually, *Pollard's function* is not an optimal choice. Therefore there is suggestion to Partition the points into about ٢٠ sets of equal size $S_{١}, \ldots, S_{٢٠}$ and that the iteration function be

$$f(Z) = \left\{ \begin{array}{lll} Z + c_1 P + d_1 Q & \text{if} & Z \in S_1 \\ Z + c_1 P + d_1 Q & \text{if} & Z \in S_2 \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ Z + c_1 P + d_1 Q & \text{if} & Z \in S_{20}. \end{array} \right.$$

Where the c_i and d_i are random integers between ١ and n – ١. The use of this iteration function gives a running time very close to that expected by theoretical estimates. In order to make computation of the values A_i and B_i more efficient, the [٥] suggest that constants $c_{١١}, \ldots, c_{٢٠}$ and $d_{١}, \ldots, d_{١٠}$ could be ***zero*** so that only one of the values A_i or B_i needs to be updated at each stage.

Example (٣.٨)

What is the discrete logarithm of $Q=(٩,\ ١٦)$ to the base $P=(٢,\ ١٨)$ in the EC $E : y^{٢} = x^{٣} + ٢٢x + ١٩$ over $F_{٢٣}$. With $\#E = ١٧ = n?$

Solution

The points of E are

$\{O, (٢,\ ٥), (٢,\ ١٨), (٥,\ ١), (٥,\ ٢٢), (٩,\ ٧), (٩,\ ١٦), (١٣,\ ٨), (١٣,\ ١٥), (١٤,\ ٩), (١٤,\ ١٤), (١٧,\ ٤), (١٧,\ ١٩), (٢٠,\ ٨), (٢٠,\ ١٥), (٢١,\ ٦), (٢١,\ ١٧)\}$

let $S_{١} = \{O, (٢,\ ٥), (٢,\ ١٨), (٥,\ ١), (٥,\ ٢٢), (٩,\ ٧)\},$

and $S_{٢} = \{(٩,\ ١٦), (١٣,\ ٨), (١٣,\ ١٥), (١٤,\ ٩), (١٤,\ ١٤), (١٧,\ ٤)\}$

and $S_{٣} = \{(١٧,\ ١٩), (٢٠,\ ٨), (٢٠,\ ١٥), (٢١,\ ٦), (٢١,\ ١٧)\}.$

Define $f(Z)$ as follows:

$$f(Z)=\begin{cases} 2Z & if \quad Z\in S_1 \\ Z+P & if \quad Z\in S_2 \\ Z+Q & if \quad Z\in S_3 . \end{cases}$$

choose $A_٠ ,B_٠ \in [١, n-١]$, let $A_٠ = ٥, B_٠ = ١٢$

and compute $Z_٠ = A_٠P + B_٠Q = ٥ * (٢, ١٨) + ١٢ * (٩, ١٦)$

$= (١٣, ٨) + (١٤, ٩)$

$= (٢٠, ٨),$

compute $Z_١ = f(Z_٠)$, since $Z_٠ \in S_٣$ then $f(Z_٠) = Z_٠ + Q$

$= (٢٠, ٨) + (٩, ١٦)$

$= (٢٠, ١٥)$

$(Z_١ , A_١ , B_١) = ((٢٠ , ١٥) , ٥, ١٣)$

since $Z_٠ \neq Z_١$

then compute $Z_٢ = f(Z_١)$, since $Z_١ \in S_٣$ then $f(Z_١) = Z_١ + Q$

$= (٢٠, ١٥) + (٩, ١٦)$

$= (٢١, ٦)$

$(Z_٢ , A_٢ , B_٢) = ((٢١ , ٦) , ٥, ١٤)$

since $Z_٠ \neq Z_١ \neq Z_٢$

then compute $Z_٣ = f(Z_٢)$, since $Z_٢ \in S_٣$ then $f(Z_٢) = Z_٢ + Q$

$= (٢١, ٦) + (٩, ١٦)$

$= (٢, ٥)$

$(Z_٣ , A_٣ , B_٣) = ((٢, ٥) , ٥, ١٥)$

since $Z_٠ \neq Z_١ \neq Z_٢ \neq Z_٣$

then compute $Z_٤ = f(Z_٣)$, since $Z_٣ \in S_١$ then $f(Z_٣) = ٢ * Z_٣$

$= ٢ * (٢, ٥)$

$= (١٤, ١٤)$

$(Z_٤ , A_٤ , B_٤) = ((١٤ , ١٤) , ١٠ , ١٣)$

since $Z_٠ \neq Z_١ \neq Z_٢ \neq Z_٣ \neq Z_٤$

then compute $Z_٥ = f(Z_٤)$, since $Z_٤ \in S_٢$ then $f(Z_٤) = Z_٤ + P$

$= (١٤, ١٤) + (٢, ١٨)$

$= (٢, ٥)$

$(Z_٥ , A_٥ , B_٥) = ((٢, ٥) , ١١ , ١٣)$

since $Z_٣ = Z_٥ = (٢, ٥)$ then we have $A_٣P + B_٣Q = A_٥P + B_٥Q$

$٥ * (٢, ١٨) + ١٥ * (٩, ١٦) = ١١ * (٢, ١٨) + ١٣ * (٩, ١٦)$

which gives $l = \dfrac{A_3 - A_5}{B_5 - B_3} mod\ 17$

$$= \frac{5-11}{13-15} mod\ 17$$

$$= \frac{6}{2} mod\ 17$$

= ٣ ***mod*** ١٧

= ٣ then the discrete logarithm of Q to the base P is ***three***.

In the above example there is another discrete logarithm of Q to the base P , { ٢٠} and this method couldn't find it, in any way there is no problem because the all ***ECDLP*** have the same result.

(٣.٨.٣) Pohlig –Hellman Attack

The ***Pohlig*** and ***Hellman*** attack reduces the problem of recovering l to the problem of recovering l modulo each of the prime factors of n = $\#E$; the desired number l can then be recovered by using the ***Chinese Remainder Theorem*** [٣٥]. We can illustrate this attack by the following algorithm [٤٢]:

- Let P and Q are two publicly points on an EC E such that $P = x * Q$, and x is integer number.
- Compute all primes factor for $\# E$ and order it as follows:

  $\# E = p_1^{\alpha_1} . p_2^{\alpha_2} ... p_k^{\alpha_k}$
- Compute the points $t_{p_i}, j = j . (\#E/p_i) . P$ such that ٠ $\leq j < p_i$, and order all results in table.
- Compute the discrete logarithm of $x \bmod p_i^{\alpha_i}$ as follows:

  $x \bmod p_i^{\alpha_i} = x_0 + x_1 p_1 + ... + x_{\alpha_i - 1} p_i^{\alpha_i - 1}$ where ٠ $\leq x_n \leq p_i -$ ١.
- To find x٠, we compute $(\#E/p_i)$ = r_{p_i}, j and compare the value of r_{p_i}, j with the table in setup ٢, we get the value of $x \bmod p_i$ when $\alpha_i =$ ١. if $\alpha_i >$ ١ the we continue to compute the value of x from $x_0, x_1, ..., x_{\alpha_i}$.
- To find x١, we compute $R = Q - x$٠$. P$, and then x١ = $(\#E/ p_i^2) * R$, if $\alpha_i =$ ٢ ,then $x = x$٠ + x١p_i , otherwise we will continue .
- Compute x٢ , at first compute $G = Q - x$٠$. P - x$١R, and then x٢ = $(\#E/ p_i^3) * G$, and in the same order we continue to compute the value required
- Using the Chinese Remainder Theorem to compute the discrete logarithm x.

Note that this method uses when we could factor n, and we couldn't use this method (the method don't work) when n is prime number

Example (٣. ٩)

What is the discrete logarithm of $Q = (٧٥,\ ١٣١)$ to the base $P = (١٨٤,\ ١٢٠)$ in the EC $E : y^{٢} = x^{٣} + ١٨٣x + ١٧٨$ over $F_{١٩٣}$. with $\# E = ١٩٦$.

Solution

$\# E = ١٩٦ = ٢^{٢} * ٧^{٢}$, then $p_{١} = ٢, p_{٢} = ٧$

$t_{p_i}, j = j\ .\ (\#E/p_i)\ .\ P$ such that $٠ \le j < p_i$

$t_{p_1}, j = j\ .\ (\#E/p_{١})\ .\ P$ such that $٠ \le j < p_{١}$,

then we must compute $t_2,0$ and $t_2,1$

$t_2,0 = ٠ * (١٩٦/٢) * (١٨٤,\ ١٢٠) = O,$

$t_2,1 = ١ * (١٩٦/٢) * (١٨٤,\ ١٢٠) = ٩٨ * (١٨٤,\ ١٢٠)$

$= O.$

$t_{p_2}, j = j\ .\ (\#E/p_{٢})\ .\ P$ such that $٠ \le j < p_{٢}$,

then we must compute $t_7,0$, $t_7,1$, $t_7,2$, $t_7,3$, $t_7,4$, $t_7,5$ and $t_7,6$

$t_7,0 = ٠ * (١٩٦/٧) * (١٨٤,\ ١٢٠) = O,$

$t_7,1 = ١ * (١٩٦/٧) * (١٨٤,\ ١٢٠) = ٢٨ * (١٨٤,\ ١٢٠)$

$= (٥٧,\ ١٨٣)$

$t_7,2 = ٢ * (١٩٦/٧) * (١٨٤,\ ١٢٠) = ٥٦ * (١٨٤,\ ١٢٠)$

$= (٨٢,\ ١٦٧)$

$t_7,3 = ٣ * (١٩٦/٧) * (١٨٤,\ ١٢٠) = ٨٤ * (١٨٤,\ ١٢٠)$

$= (٢٦,\ ١٦٠)$

$t_7,4 = ٤ * (١٩٦/٧) * (١٨٤,\ ١٢٠) = ١١٢ * (١٨٤,\ ١٢٠)$

$= (٢٦,\ ٣٣)$

$t_7,5 = ٥ * (١٩٦/٧) * (١٨٤,\ ١٢٠) = ١٤٠ * (١٨٤,\ ١٢٠)$

$= (٨٢,\ ٢٦)$

$t_7,6 = ٦ * (١٩٦/٧) * (١٨٤,\ ١٢٠) = ١٦٨ * (١٨٤,\ ١٢٠)$

$= (٥٧,\ ١٠).$

Then we order the value of t_{p_i}, j in the following table:

p_i	j						
	٠	١	٢	٣	٤	٥	٦
٢	O	O					
٧	O	(٥٧, ١٨٣)	(٨٢, ١٦٧)	(٢٦, ١٦٠)	(٢٦, ٣٣)	(٨٢, ٢٦)	(٥٧, ١٠)

To find $x \bmod p_1^{\alpha_1}$ i.e. $x \bmod 2^2$

$x \bmod 2^2 = x_0 + 2x_1$,

to find $x_{٠}$, we compute $(\#E/p_{١}) * Q =$ ٩٨ $* ($٧٥, ١٣١$)$

$$= O = t_2,0 = t_2,1$$

Then $x_{٠} =$ ٠ or ١

to find $x_{١}$, we compute $R = Q - x_{٠}P$

when $x_{٠} =$ ٠ then $R = ($٧٥, ١٣١$)$, then $(\#E/p_1^2) * R =$ ٤٩ $* ($٧٥, ١٣١$)$ $= O$

then $x_{١} =$ ٠

or if $x_{٠} =$ ١ then $R = ($٧٥, ١٣١$) - ($١٨٤, ١٢٠$)$

$= ($٧٥, ١٣١$) + ($١٨٤, $-$١٢٠$)$

$= ($١٥١, ١٢٩$)$, then $(\#E/p_1^2) * R =$ ٤٩ $* ($١٥١, ١٢٩$)=O$

then $x_{١} =$ ٠

then $x \equiv$ ١ mod ٤ or $x \equiv$ ٠ mod ٤

To find $x\,mod\,p_2^{\alpha_2}$ i.e. $x\,mod\,7^2$

$x\,mod\,7^2 = x_0 + 7x_1$,

to find $x_{٠}$, we compute $(\#E/p_{٢}) * Q =$ ٢٨ $* ($٧٥, ١٣١$)$

$= ($٨٢, ٢٦$) = t_7,5$, then $x_{٠} =$ ٥

to find $x_{١}$, we compute $R = Q - x_{٠}.P = ($٧٥, ١٣١$) -$ ٥ $* ($١٨٤, ١٢٠$)$

$= ($٧٥, ١٣١$) + ($١١٧, $-$١١٩$)$

$= ($٥٧, ١٠$)$

$(\#E/p_2^2) * R =$ ٤ $* ($٥٧, ١٠$) = ($٢٦, ١٦٠$) = t_7,3$ then $x_{١} =$ ٣

$x \equiv$ ٥ $+$ ٧ $*$ ٣ mod ٤٩

$x \equiv$ ٢٦ mod ٤٩,

Now we will use the Chinese Remainder Theorem to solve the system of equations: $\begin{Bmatrix} x \equiv 1 mod\ 4 \\ x \equiv 26\ mod\ 49 \end{Bmatrix}$ or $\begin{Bmatrix} x \equiv 0\ mod\ 4 \\ x \equiv 26\ mod\ 49 \end{Bmatrix}$

to solve $\begin{cases} x \equiv 1 mod\ 4 \\ x \equiv 26\ mod\ 49 \end{cases}$

$n = n_{١} * n_{٢}$
$=$ ٤ $*$ ٤٩
$=$ ١٩٦.

$N_{١} = n / n_{١}$
$=$ ١٩٦/ ٤
$=$ ٤٩, $\rightarrow$ ٤٩$x_{١} \equiv$ ١ $(mod$ ٤$) \rightarrow x_{١} =$ ١,

$N_{٢} = n / n_{٢}$,
$=$ ١٩٦/ ٤٩,
$=$ ٤, $\rightarrow$ ٤$x_{٢} \equiv$ ١ $(mod$ ٤٩$) \rightarrow x_{٢} =$ ٣٧,

but $\bar{x} = (a_{١} * N_{١} * x_{١} + a_{٢} * N_{٢} * x_{٢})\ mod\ n,$

so $\bar{x} = (١ * ٤٩ * ١ + ٢٦ * ٤ * ٣٧) \bmod ١٩٦$

$\bar{x} = ١٧٣.$

So either the discrete logarithm of Q to the base P is ١٧٣.

To solve $\left\{ \begin{array}{c} x \equiv 0 \bmod 4 \\ x \equiv 26 \bmod 49 \end{array} \right\}$

$$n = n_١ * n_٢$$
$$= ٤ * ٤٩$$
$$= ١٩٦.$$
$$N_١ = n / n_١$$
$$= ١٩٦ / ٤$$
$$= ٤٩, \quad \rightarrow ٤٩x_١ \equiv ١ \pmod{٤} \rightarrow x_١ = ١,$$
$$N_٢ = n / n_٢,$$
$$= ١٩٦ / ٤٩,$$
$$= ٤, \quad \rightarrow ٤x_٢ \equiv ١ \pmod{٤٩} \rightarrow x_٢ = ٣٧,$$

but $\bar{x} = (a_١ * N_١ * x_١ + a_٢ * N_٢ * x_٢) \bmod n,$

so $\bar{x} = (٠ * ٤٩ * ١ + ٢٦ * ٤ * ٣٧) \bmod ١٩٦$

$\bar{x} = ١٢٤.$

Alternatively, the discrete logarithm of Q to the base P is ١٢٤.

In the above example there are another discrete logarithm of Q to the base P, { ٢٦, ٧٥} and this method couldn't find it, in any way there is no problem because the all ***ECDLP*** have the same result.

After all these, the breakers of the ciphering can find the secret key which can be used in any cryptosystem , and compute the inverse of it, and then fine the decipher text, but when the p has strength bits, then the solution of the ***ECDLP*** becomes more difficult (as we will see in the section ***(٣. ١٠)***).

(٣. ٩) Choosing a Suitable Elliptic Curve for Cryptosystem

After reviewing the attacks we have mentioned, it should be apparent that the choice of the EC E and its underlying field K has enormous impact on the speed, efficiency, key length (i.e. practicality) and security of any ECCs. Although E, K and a base point $P \in E$ are all fixed and publicly known prior to the encryption process, the task of selecting them for a given scheme is the most important step. We will explore some of the choices here [٧].

- The two most common choices in practical applications for the underlying finite field are $F_٢^m$ and F_p (where p is an odd prime). The ***ECDLP*** is equally difficult for instances which use $F_{٢^m}$ as those which use F_p, and where the sizes ٢^m and ***p*** of the fields are approximately equal [٦٣],[٣٣]. There have not been any mathematical discoveries to date which suggest that the ***ECDLP*** for ECs over $F_{٢^m}$ may be any easier or harder than the ***ECDLP*** for ECs over F_p [٣٣].

- To choose the “right” EC, we first need to know what kind of curve we want and what types we can use. There are infinite varieties of ECs to choose from but a selecting few have been of interest to the study of ECCss [٧]. In the following, we shall present two classes of ECs that have been used in various encryption schemes.
  - ***Supersingular Curves***: ***Menezes*** and ***Vanstone*** have examined the advantages of supersingular ECs in cryptosystems, specifically those over the field $F_{٢^r}$. An EC over a finite field of ***q*** elements is said to be supersingular if $t^٢ =$ ٠*, q,* ٢*q,* ٣*q* or ٤*q* where t is defined as $t = q +$ ١ – ***#E(Fq)***; $|t| \leq 2\sqrt{q}$ [٧]. An EC over a field of characteristic ٢ or ٣ is supersingular if and only if it has a ***zero j-invariant*** [٥٣], [٧]. For example $y^٢ + a_٣y = x^٣ + a_٤x + a_٦$ is a supersingular curve.
  - ***Nonsupersingular Curves:*** A ***nonsupersingular*** curve or an “***ordinary***” EC has a ***nonzero j-invariant*** [٥٣],[٧]. For example, an EC defined by equation in ***(٢.٥.١.٣)*** is a nonsupersingular curve. This curve is used for cryptography.

The advantage that a nonsupersingular curve has over a supersingular curve is that it can provide the same level of security as the supersingular curve, but with a much smaller underlying field [٧].

After these, for large p we can apply the ***Koblitz's Random Selection Method*** [٧]:

- *Randomly select three elements from F_p; call them $x, y, a.$
- Set the value for b by computing $b = y^٢ - (x^٣ + ax)$ since equation of the EC over F_p is $y^٢ = x^٣ + ax + b.$
- Check that the cubic on the right side of the above equation does not have multiple roots, i.e. check that $٤a^٣ + ٢٧b^٢ \neq ٠$.
- If the previous condition is not met, return to step *****.
- Else set $P = (x, y)$ (preferably, be base point but not necessarily [٤٣]) and let $y^٢ = x^٣ + ax + b$ be our EC

(٣.١٠) Security of Elliptic Curve Cryptosystem

The cryptographic strength of EC encryption lies in the difficulty for a cryptanalyst to determine the secret random number ***k*** from ***kP*** and ***P*** itself [١٥], i.e. the difficult of solve the ECDLP when ***p*** has very large bits. For more sure a ١٠٢٤– ***bit*** RSA (which is one of the more widely uses in cryptography) key has approximately the same strength as a ١٦٠– ***bit*** ECC key [٤١],[٣٠],[٢٨],[٣٣], and it is estimated that it should take ١٠^{١١} Million Instructions Per Second (MIPS) years to break a key of that strength. In addition, ٢٠٤٨– ***bit*** RSA key corresponds to a ٢١٠– ***bit*** ECC key, and the estimated time to break these is ١٠^{٢٠} MIPS years [٣٠]. This means that we can use shorter keys (compared to other cryptosystems) for high security levels [١٣].

Chapter Four

Developments of Elliptic Curves Cryptosystem

(٤. ١) Introduction

ECCs have the potential to provide relatively small block size, high-security public key schemes that can be efficiently implemented [٣٢].

When we study the ElGamal ECCs, we saw that if the sender went to send any message to the receiver, the first must use the public key of second (as the other public key cryptosystem), and in way cannot be developed, but we can vary it with the same complexity as in *(٤. ٢. ٢)*, because the addition and subtraction have the same computation complexity.

However, in the ***MVECC*** which is a very important system of a public key cryptosystem we have the following:

- It dose not depend on additive operation on EC group,
- The message needs not to be a point on EC.

Therefore we can use this to develop the encryption and decryption scheme, as in the proposition *(٤. ٢. ٣. ١)*, *(٤. ٢. ٣. ٢)*, and *(٤. ٢. ٣. ٣)*, with more complexity even the complexity of propositions in [٣٤] in ٢٠٠٤.

We also try to benefit from the DHEK to use this key (the key come from DHEK algorithm) as a secret key in tow suggestion methods.

Finally we listed in this chapter all software which we used in our working. All these software are written in MATLAB (The Language of Technical Computing) , version ٦.٥ .

Now we start this chapter by mentioning the four propositions methods[١] for enhancing of ECCs which coming in [٣٤].

Theorem (٤. ١)

Given the pair $(k_{١}, k_{٢})$ and the message $(m_{١}, m_{٢})$ such that $k_{١}, k_{٢}, m_{١}, m_{٢}$ in the field F_p and let

$$c_{١} = (m_{١}k_{١} + m_{٢}k_{٢}) \bmod p$$
$$c_{٢} = (m_{١}k_{١} - m_{٢}k_{٢}) \bmod p$$

Then

$$m_{١} = (c_{١} + c_{٢})(٢k_{١})^{-١} \bmod p$$
$$m_{٢} = (c_{١} - c_{٢})(٢k_{٢})^{-١} \bmod p.$$

Theorem (٤. ٢)

Given the pair $(k_{١}, k_{٢})$ and the message $(m_{١}, m_{٢})$ such that $k_{١}, k_{٢}, m_{١}, m_{٢}$ in the field F_p and let

$$c_{١} = (m_{١}k_{١} + m_{٢}k_{٢}) \bmod p$$
$$c_{٢} = (m_{١}k_{١} - m_{٢}k_{٢}) \bmod p$$

Then

$$m_{١} = ((c_{١} + c_{٢})k_{٢})(٢k_{٢}k_{١})^{-١} \bmod p$$
$$m_{٢} = ((c_{١} - c_{٢})k_{١})(٢k_{٢}k_{١})^{-١} \bmod p.$$

Theorem (٤. ٣)

Given the pair $(k_{١}, k_{٢})$ and the message $(m_{١}, m_{٢})$ such that $k_{١}, k_{٢}, m_{١}, m_{٢}$ in the field F_p and let

$$c_{١} = (m_{١}k_{١} - m_{٢}) \bmod p$$
$$c_{٢} = (m_{١} - m_{١}k_{١}k_{٢} + m_{٢}k_{٢}) \bmod p$$

Then

$$m_{١} = (c_{١}k_{٢} + c_{٢}) \bmod p$$
$$m_{٢} = (c_{١}k_{١}k_{٢} - c_{١} + c_{٢}k_{١}) \bmod p.$$

Theorem (٤. ٣)

Given the pair $(k_{١}, k_{٢})$ and the message $(m_{١}, m_{٢})$ such that $k_{١}, k_{٢}, m_{١}, m_{٢}$ in the field F_p and let

$$c_{١} = m_{١}\, k_1^{k_2} \bmod p$$
$$c_{٢} = m_{٢}\, k_2^{k_1} \bmod p$$

Then

$$m_{١} = c_{١}\, k_1^{(-k_2 \bmod p-1)} \bmod p$$
$$m_{٢} = c_{٢}\, k_2^{(-k_1 \bmod p-1)} \bmod p.$$

[١] These propositions methods coming as theorems.

(٤. ٢)The Proposed Developments

(٤. ٢. ١) Introduction

In this subsection, we present the proposition and suggestion methods to encryption–decryption by group of EC, depending on ElGamal, MVECC and DHEK. Finally, we listed in this chapter all software which we used in our working. Now, we first will vary ElGamal ECCs with same complexity.

(٤. ٢. ٢) Proposition to Variant ElGamal Elliptic Curve Cryptosystem

To vary the encryption and decryption of ElGamal ECCs. Let $E(F)$ be an EC group and let B be a base point on E . The user Benin first selects a private key d and generates a public key $Q = d\,B$.

If Ali likes to encrypt and send a message M to Benin, he should choose a random positive integer e and produce the ciphertext C_M , such that $C_M =\{C , e\,B\}$

Where $C = M - e\,Q$

To decrypt the ciphertext, Benin computes the following:

$$\begin{aligned} C + d\,(e\,B) &= M - e\,Q + d\,(e\,B) \\ &= M - e\,(d\,B) + d\,(e\,B) \\ &= M. \end{aligned}$$

Example (٤. ١)

Let EC E defined over F_p (p = ٧٢١٦٩ with parameters a = ٧١٦٦٩, b = ٧١٤٧٠ where (٤$a^{٣}$+ ٢٧ $b^{٢}$)$mod\ p$ = ٤٤٣٠١ ≠ ٠). Suppose the private key of Benin is d = ٦٢٤٣, and the private key of Ali is e = ٤٧٨١.and let B= (٧١٨٢٥, ٧١٨٦١) be a base point on E, if M = (٧٢١١٦, ٧١٤٩٥) is the message point, discuss what Ali and Benin should do if Ail want to send M to Benin.

Solution:

Since d = ٦٢٤٣ thus the public key of Benin is

Q = ٦٢٤٣(٧١٨٢٥, ٧١٨٦١) = (٣٨٢١٦, ٤٧٥١)

If Ali wishes to a message to Benin he should do the following:

- Compute eQ = ٤٧٨١ (٣٨٢١٦, ٤٧٥١) = (٥٩٥٨٣, ٥٥٧٠٣)
- Compute eB = ٤٧٨١ (٧١٨٢٥, ٧١٨٦١) = (٥٦١٣٠, ٢١٤٥٨)

–Compute C:

$C = M – e\ Q$ = (٧٢١١٦, ٧١٤٩٥)–(٥٩٥٨٣, ٥٥٧٠٣)
= (٧٢١١٦, ٧١٤٩٥) + (٥٩٥٨٣, – ٥٥٧٠٣)
= (٩٢٥, ٥٩٤٦٨)

Then Ali send $C_M = \{C,e\ B\}$
= {(٩٢٥, ٥٩٤٦٨),(٥٦١٣٠, ٢١٤٥٨)} to Benin.

To decrypt the ciphertext, Benin does the following:

–Compute $d\ (e\ B)$ = ٦٢٤٣(٥٦١٣٠, ٢١٤٥٨)
= (٥٩٥٨٣, ٥٥٧٠٣)

–Compute M:

$M = C + d\ (e\ B)$
= (٩٢٥, ٥٩٤٦٨) + (٥٩٥٨٣, ٥٥٧٠٣)
= (٧٢١١٦, ٧١٤٩٥)
= M.

Example (٤. ٢)

Let EC E defined over F_p (p = ١٠٥٥٥٧ with parameters a = ١١١١, b = ٢٢٢٤ where (٤$a^{٣}$+ ٢٧$b^{٢}$)$\mathrm{mod}\ p$ = ١٠٠٢١ ≠ ٠). Suppose the private key of Benin is d = ٩٨١٢٤, and the private key of Ali is e = ٤٧٩١٣.and let B= (١٠٥٢٥٨, ٣٣٦٢١) be a base point on E, if M = (١٠٥٢٧٢, ٩٧٠٩٩) is the message point, discuss what Ali and Benin should do if Ail want to send M to Benin.

Solution:

Since d = ٩٨١٢٤ thus the public key of Benin is

Q = ٩٨١٢٤ (١٠٥٢٥٨, ٣٣٦٢١) = (٤١٤٨٣, ٢١٨٠٦)

If Ali likes to send a message to Benin, then he should do the following:

- Compute eQ = ٤٧٩١٣(٤١٤٨٣, ٢١٨٠٦) = (٦٨٠٨٦, ٧٦٦٤٨)
- Compute eB = ٤٧٩١٣(١٠٥٢٥٨, ٣٣٦٢١) = (٧٠٤٣١,٥٠٤٢١)

–Compute C:

$C = M – e\ Q$ = (١٠٥٢٧٢, ٩٧٠٩٩)–(٦٨٠٨٦, ٧٦٦٤٨)
= (١٠٥٢٧٢, ٩٧٠٩٩) + (٦٨٠٨٦,– ٧٦٦٤٨)
= (٩٢٤٧٥, ٦١٥٠٣)

Then Ali send $C_M = \{C, e\ B\}$
= {(٩٢٤٧٥, ٦١٥٠٣),(٧٠٤٣١,٥٠٤٢١)} to Benin.

To decrypt the ciphertext, Benin does the following:

–Compute $d\ (e\ B)$ = ٩٨١٢٤ (٧٠٤٣١,٥٠٤٢١)
= (٦٨٠٨٦,٧٦٦٤٨)

–Compute M:

$$\begin{aligned} M &= C + d\,(e\,B) \\ &= (٩٢٤٧٥,\ ٦١٥٠٣) + (٦٨٠٨٦, ٧٦٦٤٨) \\ &= (١٠٥٢٧٢, ٩٧٠٩٩) \\ &= M. \end{aligned}$$

(٤. ٢. ٣) Proposition to Development of MVECC

The development of the encryption and decryption of MVECC is as follows:

(٤. ٢. ٣. ١) Suppose Ali wants to send a message $M = (m_{١}, m_{٢})$ to Benin,
Let d denote Benin's secret key and $Q = d\ B$ [B is a point on E] denote Benin's public key . Ali chooses a random integer e and sends C_M:

$$C_M = \{C, eB\}$$

Where $C = (c_{١}, c_{٢})$

$$(k_{١},k_{٢}) = eQ$$

$$c_{١} = (m_{١} + k_{١}\,k_{٢}) \bmod p$$

$$c_{٢} = m_{١}\,(m_{٢} + k_{٢}\,k_{١}) \bmod p$$

To decrypt the ciphertext Benin computes:

$$(k_{١}, k_{٢}) = d\,(e\,B)$$

$$m_{١} = (c_{١} - k_{١}k_{٢}) \bmod p$$

$$m_{٢} = (m_1^{-1}\,c_{٢} - k_{١}\,k_{٢}) \bmod p$$

Proof:

$$(c_{١} - k_{١}k_{٢}) \bmod p = (m_{١} + k_{١}\,k_{٢} - k_{١}k_{٢}) \bmod p$$
$$= m_{١}.$$

$$(m_1^{-1}\,c_{٢} - k_{١}\,k_{٢}) \bmod p = (m_1^{-1}\,m_{١}\,(m_{٢} + k_{٢}\,k_{١}) - k_{١}k_{٢}) \bmod p$$
$$= m_{٢}.$$

(٤. ٢. ٣. ٢) Suppose Ali wants to sent a message $M = (m_{١}, m_{٢})$ to Benin, Let d denotes Benin's secret key and $Q = dB$ (B is a point on E) denotes Benin's public key. Ali chooses a random integer e and sends C_M:

$$C_M = \{C, eB\}$$

Where $C = (c_{١}, c_{٢})$

$$(k_{١}, k_{٢}) = e\,Q$$

$$c_{١} = (m_{١} * (k_{١}\,k_{٢} - k_{١})) \bmod p$$

$$c_{٢} = (m_{٢} * (k_{١}\,k_{٢} - k_{٢})) \bmod p$$

To decrypt the ciphertext Benin computes:

$$(k_{١}, k_{٢}) = d\,(e\,B)$$

$$m_{١} = (c_{١} * (k_{١}\,k_{٢} - k_{١})^{-١}) \bmod p$$

$$m_٢ = (c_٢ * (k_١ k_٢ - k_٢)^{-١}) \bmod p$$

Proof:

$$(c_١ * (k_١ k_٢ - k_١)^{-١}) \bmod p = (m_١ * (k_١ k_٢ - k_١) * (k_١ k_٢ - k_١)^{-١}) \bmod p$$
$$= m_١ .$$
$$(c_٢ * (k_١ k_٢ - k_٢)^{-١}) \bmod p = (m_٢ * (k_١ k_٢ - k_٢) * (k_١ k_٢ - k_٢)^{-١}) \bmod p$$
$$= m_٢ .$$

(٤.٢.٣.٣) Suppose Ali wants to send a message $M = (m_١, m_٢)$ to Benin,

Let d denotes Benin's secret key and $Q=dB$ [B is a point on E] denotes Benin's public key. Ali chooses a random integer e and sends C_M:

$$C_M = \{C, eB\}$$

where $C = (c_١, c_٢)$

$$(k_١, k_٢) = e\,q$$
$$c_١ = m_١ + (k_1 k_2^{k_1})^{-1} \bmod p$$
$$c_٢ = m_٢ + (k_2 k_1^{k2})^{-1} \quad \bmod p$$

To decrypt the ciphertext Benin computes:

$$(k_١, k_٢) = d\,(e\,B)$$
$$m_١ = (c_١ - (k_1 k_2^{k_1})^{-1}) \bmod p$$
$$m_٢ = (c_٢ - (k_2 k_1^{k_2})^{-1}) \bmod p$$

Proof:

$$(c_١ - (k_1 k_2^{k_1})^{-1}) \bmod p = (m_١ + (k_1 k_2^{k_1})^{-1} - (k_1 k_2^{k_1})^{-1}) \bmod p$$
$$= m_١ .$$
$$(c_٢ - (k_2 k_1^{k_2})^{-1}) \bmod p = (m_٢ + (k_2 k_1^{k_2})^{-1} - (k_2 k_1^{k_2})^{-1}) \bmod p$$
$$= m_٢ .$$

Example(٤.٣):

Let EC E defined over F_p (p = ٧٧١٧١ with parameters a = ٧٦٦٧١, b = ٧٦٦٧١ where $(٤a^٣ + ٢٧b^٢) \bmod p = ٢٧٠٣٢ \neq ٠$.
Suppose the private key of Benin is d = ٧١٤٢, then the public key of Benin is $Q = d\,B$ (B = (٧٦٨٢٥, ٧٦٩٧٨) is a base point on E)

$$\therefore Q = ٧١٤٢(٧٦٨٢٥, ٧٦٩٧٨)$$
$$= (١٥٧٧٤, ٣٩٦٢٥)$$

and the private key of Ali is e = ٣٦٧١٢.

Solution:

- Using <u>(٤.٢.٣.١)</u> method : If Ali would like to send a message $M = (\text{٦٣٥١٢}, \text{٤٢٩٣}) = (m_{\text{١}}, m_{\text{٢}})$ to Benin, then he should do the following:

–Compute $e\,Q = \text{٣٦٧١٢}(\text{١٥٧٧٤}, \text{٣٩٦٢٥})$

$= (\text{٣٨٨٣٤}, \text{٨٨٤١}) = (k_{\text{١}}, k_{\text{٢}})$

–Compute $e\,B = \text{٣٦٧١٢}(\text{٧٦٨٢٥}, \text{٧٦٩٧٨}) = (\text{٢٢٧٢١}, \text{١٩٥٥٣})$

–Compute C:

$$C = (c_{\text{١}}, c_{\text{٢}})$$

$c_{\text{١}} = m_{\text{١}} + k_{\text{١}} k_{\text{٢}} \bmod p$

$= \text{٦٣٥١٢} + \text{٣٨٨٣٤} * \text{٨٨٤١} \bmod \text{٧٧١٧١}$

$= \text{٦١١٢٧}$

$c_{\text{٢}} = m_{\text{١}}(m_{\text{٢}} + k_{\text{٢}} k_{\text{١}}) \bmod p$

$= \text{٦٣٥١٢}(\text{٤٢٩٣} + \text{٣٨٨٣٤} * \text{٨٨٤١}) \bmod \text{٧٧١٧١}$

$= \text{٢٢٤٢٦}$

Then Ali sends $C_M = \{C, e\,B\}$

$= \{(\text{٦١١٢٧}, \text{٢٢٤٢٦}), (\text{٢٢٧٢١}, \text{١٩٥٥٣})\}$ to Benin.

To decrypt the ciphertext , Benin should do the following:

–Compute $d\,(e\,B) = \text{٧١٤٢}(\text{٢٢٧٢١}, \text{١٩٥٥٣})$

$= (\text{٣٨٨٣٤}, \text{٨٨٤١}) = (k_{\text{١}}, k_{\text{٢}})$

–Compute M:

$$M = (m_{\text{١}}, m_{\text{٢}})$$

$m_{\text{١}} = c_{\text{١}} - k_{\text{١}} k_{\text{٢}} \bmod p$

$= \text{٦١١٢٧} - \text{٣٨٨٣٤} * \text{٨٨٤١} \bmod \text{٧٧١٧١}$

$= \text{٦٣٥١٢}$

$= m_{\text{١}}$

$m_{\text{٢}} = (m_1^{-1} c_{\text{٢}} - k_{\text{١}} k_{\text{٢}}) \bmod p$

$m_1^{-1} = (\text{٦٣٥١٢})^{-\text{١}} \bmod \text{٧٧١٧١}$

$= \text{٤١٠٤٦}$

then $m_{\text{٢}} = (\text{٤١٠٤٦} * \text{٢٢٤٢٦} - \text{٣٨٨٣٤} * \text{٨٨٤١}) \bmod \text{٧٧١٧١}$

$= \text{٤٢٩٣}$

$= m_{\text{٢}}$

- Using <u>(٤.٢.٣.٢)</u> method : If Ali would like to send a message $M = (\text{٦٣٥١٢}, \text{٤٢٩٣}) = (m_{\text{١}}, m_{\text{٢}})$ to Benin, then he should do the following:

–Compute $e\,Q = \text{٣٦٧١٢}(\text{١٥٧٧٤}, \text{٣٩٦٢٥})$

$= (\text{٣٨٨٣٤}, \text{٨٨٤١}) = (k_{\text{١}}, k_{\text{٢}})$

–Compute $e\,B = \text{٣٦٧١٢}(\text{٧٦٨٢٥}, \text{٧٦٩٧٨}) = (\text{٢٢٧٢١}, \text{١٩٥٥٣})$

–Compute C:

$$C = (c_{\text{١}}, c_{\text{٢}})$$

$c_{\text{١}} = (m_{\text{١}} * (k_{\text{١}} k_{\text{٢}} - k_{\text{١}})) \bmod p$

$= (\text{٦٣٥١٢} * (\text{٣٨٨٣٤} * \text{٨٨٤١} - \text{٣٨٨٣٤})) \bmod \text{٧٧١٧١}$

$= \text{٤٧٨٧٦}$

$c_{\text{٢}} = (m_{\text{٢}} * (k_{\text{١}}\, k_{\text{٢}} - k_{\text{٢}}))\bmod p$

$= (\text{٤٢٩٣}*(\text{٣٨٨٣٤}* \text{٨٨٤١} - \text{٨٨٤١}))\bmod \text{٧٧١٧١}$

$= \text{٣٨٦٥٧}$

Then Ali sends $C_M = \{C, e\,B\}$

$= \{(\text{٤٧٨٧٦}, \text{٣٨٦٥٧}), (\text{٢٢٧٢١}, \text{١٩٥٥٣})\}$ to Benin.

To decrypt the ciphertext, Benin should do the following:

–Compute $d\,(e\,B) = \text{٧١٤٢}(\text{٢٢٧٢١}, \text{١٩٥٥٣})$

$= (\text{٣٨٨٣٤}, \text{٨٨٤١}) = (k_{\text{١}}, k_{\text{٢}})$

–Compute M:

$M = (m_{\text{١}}, m_{\text{٢}})$

$(k_{\text{١}}\, k_{\text{٢}} - k_{\text{١}})^{-\text{١}} \bmod p = (\text{٣٨٨٣٤}* \text{٨٨٤١} - \text{٣٨٨٣٤})^{-\text{١}} \bmod \text{٧٧١٧١}$

$= \text{٣٤٣٢٩٢٥٦٠}^{-\text{١}} \bmod \text{٧٧١٧١}$

$= \text{٤٦٩٥٩}$

$m_{\text{١}} = (c_{\text{١}} *(k_{\text{١}}\, k_{\text{٢}} - k_{\text{١}})^{-\text{١}}) \bmod p$

$= \text{٤٧٨٧٦} * \text{٤٦٩٥٩} \bmod \text{٧٧١٧١}$

$= \text{٦٣٥١٢}$

$= m_{\text{١}}.$

$(k_{\text{١}}\, k_{\text{٢}} - k_{\text{٢}})^{-\text{١}} \bmod p = (\text{٣٨٨٣٤}* \text{٨٨٤١} - \text{٨٨٤١})^{-\text{١}} \bmod \text{٧٧١٧١}$

$= \text{٦٥٩٤٥}^{-\text{١}} \bmod \text{١٧}$

$= \text{٤١٠٧٤}$

$m_{\text{٢}} = (c_{\text{٢}} * (k_{\text{١}}\, k_{\text{٢}} - k_{\text{٢}})^{-\text{١}}) \bmod p$

$= \text{٣٨٦٥٧} * \text{٤١٠٧٤} \bmod \text{٧٧١٧١}$

$= \text{٤٢٩٣}$

$= m_{\text{٢}}.$

- Using (٤.٢.٣.٣) method : If Ali wishes to send a message $M = (\text{٦٣٥١٢}, \text{٤٢٩٣}) = (m_{\text{١}}, m_{\text{٢}})$ to Benin, then he should do the following:

–Compute $e\,Q = \text{٣٦٧١٢}(\text{١٥٧٧٤}, \text{٣٩٦٢٥})$

$= (\text{٣٨٨٣٤}, \text{٨٨٤١}) = (k_{\text{١}}, k_{\text{٢}})$

–Compute $e\,B = \text{٣٦٧١٢}(\text{٧٦٨٢٥}, \text{٧٦٩٧٨}) = (\text{٢٢٧٢١}, \text{١٩٥٥٣})$

–Compute C:

$C = (c_{\text{١}}, c_{\text{٢}})$

$c_{\text{١}} = (m_{\text{١}} + (k_1\, k_2^{k_1})^{-1}) \bmod p$

$= (\text{٦٣٥١٢} + \text{١٩١٤٥}) \bmod \text{٧٧١٧١}$

$= \text{٥٤٨٦}$

$c_{\text{٢}} = (m_{\text{٢}} + (k_{\text{٢}} k_{\text{١}}{}^{k}{}_{\text{٢}})^{-\text{١}} \bmod p$

$= (\text{٤٢٩٣} + \text{٤٩٩٦٦}) \bmod \text{٧٧١٧١}$

$= \text{٥٤٢٥٩}$

Then Ali sends $C_m = \{C, eB\}$

$=\{(٥٤٨٦,\ ٥٤٢٥٩), (٢٢٧٢١,\ ١٩٥٥٣)\}$ to Benin.

To decrypt the ciphertext, Benin should do the following:

–Compute $d\ (e\ B) = ٧١٤٢(٢٢٧٢١,\ ١٩٥٥٣)$

$= (٣٨٨٣٤,\ ٨٨٤١) = (k_{١}, k_{٢})$

–Compute M:

$M = (m_{١}, m_{٢})$

$m_{١} = (c_{١} - (k_{١}k_{٢}{}^{k}{}_{١})^{-١}) \bmod p$

$= (٥٤٨٦ - ١٩١٤٥) \bmod ٧٧١٧١$

$= ٦٣٥١٢$

$= m_{١}$

$m_{٢} = (c_{٢} - (k_{٢}k_{١}{}^{k}{}_{٢})^{-١}) \bmod p$

$= (٥٤٢٥٩ - ٤٩٩٦٦) \bmod ١٧$

$= ٤٢٩٣$

$= m_{٢}.$

Example (٤.٤):

Let EC E defined over F_p (p = ١٠٥٥٥٧ with parameters $a = ١١١١$, $b = ٢٢٢٤$ where $(٤a^{٣} + ٢٧b^{٢}) \bmod p = ١٠٠٢١ \neq ٠$.
Suppose the private key of Benin is $d = ٨٥٦١١$, then the public key of Benin is $Q = d\,B$ ($B = (١٠٥٢٨٠,\ ١٢٢٢٩)$ is a base point on E)

$\therefore Q = ٨٥٦١١(١٠٥٢٨٠,\ ١٢٢٢٩)$

$= (٦٧١٥٣, ١٠١١٧)$

and the private key of Ali is $e = ٦٦٦١٢$.

Solution:

- Using (٤.٢.٣.١) method : If Ali wishes to send a message $M = (٧٢٢٣٥,\ ٤٩٥٨٣) = (m_{١}, m_{٢})$ to Benin, then he should do the following:

–Compute $e\,Q = ٦٦٦١٢(٦٧١٥٣, ١٠١١٧)$

$= (٥٣١٣٤,\ ٦٠٧٠٢)$

$= (k_{١}, k_{٢})$

–Compute $e\,B = ٦٦٦١٢(١٠٥٢٨٠,\ ١٢٢٢٩)$

$= (٨٦٣٢٨, ١٥١٨٥)$

–Compute C:

$C = (c_{١}, c_{٢})$

$c_{١} = (m_{١} + k_{١}k_{٢}) \bmod p$

$= (٧٢٢٣٥ + ٥٣١٣٤ * ٦٠٧٠٢) \bmod ١٠٥٥٥٧$

$= ١٢٦١١$

$c_{٢} = m_{١}(m_{٢} + k_{٢}k_{١}) \bmod p$

$= ٧٢٢٣٥(٤٩٥٨٣ + ٥٣١٣٤ * ٦٠٧٠٢) \bmod ١٠٥٥٥٧$

$= ٧٦٠٦٩$

Then Ali sends $C_M = \{C, e\,B\}$

$= \{(١٢٦١١,\ ٧٦٠٦٩), (٨٦٣٢٨, ١٥١٨٥)\}$ to Benin.

To decrypt the ciphertext , Benin should do the following:

–Compute $d\ (e\ B) =$ $٨٥٦١١\ (٨٦٣٢٨, ١٥١٨٥)$

$= (٥٣١٣٤,\ ٦٠٧٠٢)$

$= (k_١, k_٢)$

–Compute M:

$M = (m_١, m_٢)$

$m_١ = c_١ – k_١\, k_٢ \bmod p$

$= (١٢٦١١ – ٥٣١٣٤ * ٦٠٧٠٢) \bmod ١٠٥٥٥٧$

$= ٧٢٢٣٥$

$= m_١$

$m_٢ = (m_1^{-1} c_٢ – k_١\, k_٢) \bmod p$

$m_1^{-1} = (٧٢٢٣٥)^{-١} \bmod ١٠٥٥٥٧$

$= ١١٧٨١$

then $m_٢ = (١١٧٨١ * ٧٦٠٦٩ – ٥٣١٣٤ * ٦٠٧٠٢) \bmod ١٠٥٥٥٧$

$= ٤٩٥٨٣$

$= m_٢$

- Using (٤.٢.٣.٢) method : If Ali wishes to send a message $M = (٧٢٢٣٥,\ ٤٩٥٨٣) = (m_١, m_٢)$ to Benin, then he should do the following:

–Compute $e\ Q =$ $٦٦٦١٢(٦٧١٥٣, ١٠١١٧)$

$= (٥٣١٣٤,\ ٦٠٧٠٢)$

$= (k_١, k_٢)$

–Compute $e\ B =$ $٦٦٦١٢(١٠٥٢٨٠,\ ١٢٢٢٩)$

$= (٨٦٣٢٨, ١٥١٨٥)$

–Compute C:

$C = (c_١, c_٢)$

$c_١ = (m_١ * (k_١\, k_٢ – k_١)) \bmod p$

$= (٧٢٢٣٥ * (٥٣١٣٤ * ٦٠٧٠٢ – ٥٣١٣٤)) \bmod ١٠٥٥٥٧$

$= ٢٠٦٦١$

$c_٢ = (m_٢ * (k_١\, k_٢ – k_٢)) \bmod p$

$= (٤٩٥٨٣ * (٥٣١٣٤ * ٦٠٧٠٢ – ٦٠٧٠٢)) \bmod ١٠٥٥٥٧$

$= ٦٣١٣٩$

Then Ali sends $C_M = \{C, e\ B\}$

$= \{(٢٠٦٦١,\ ٦٣١٣٩), (٨٦٣٢٨, ١٥١٨٥)\}$ to Benin.

To decrypt the ciphertext, Benin should do the following:

–Compute $d\ (e\ B) = ٨٥٦١١\ (٨٦٣٢٨, ١٥١٨٥)$

$= (٥٣١٣٤,\ ٦٠٧٠٢)$

$= (k_١, k_٢)$

–Compute M:

$$M = (m_{\text{١}}, m_{\text{٢}})$$

$$(k_{\text{١}} k_{\text{٢}} - k_{\text{١}})^{-\text{١}} \bmod p = (\text{٥٣١٣٤} * \text{٦٠٧٠٢} - \text{٥٣١٣٤})^{-\text{١}} \bmod \text{١٠٥٥٥٧}$$
$$= \text{٩٨٣٥٦}^{-\text{١}} \bmod \text{١٠٥٥٥٧}$$
$$= \text{٩٠٩١٣}$$

$$m_{\text{١}} = (c_{\text{١}} * (k_{\text{١}} k_{\text{٢}} - k_{\text{١}})^{-\text{١}}) \bmod p$$
$$= \text{٢٠٦٦١} * \text{٩٠٩١٣} \bmod \text{١٠٥٥٥٧}$$
$$= \text{٧٢٢٣٥}$$
$$= m_{\text{١}}.$$

$$(k_{\text{١}} k_{\text{٢}} - k_{\text{٢}})^{-\text{١}} \bmod p = (\text{٥٣١٣٤} * \text{٦٠٧٠٢} - \text{٦٠٧٠٢})^{-\text{١}} \bmod \text{١٠٥٥٥٧}$$
$$= \text{٩٠٧٨٨}^{-\text{١}} \bmod \text{١٠٥٥٥٧}$$
$$= \text{٥٥٠١٢}$$

$$m_{\text{٢}} = (c_{\text{٢}} * (k_{\text{١}} k_{\text{٢}} - k_{\text{٢}})^{-\text{١}}) \bmod p$$
$$= \text{٦٣١٣٩} * \text{٥٥٠١٢} \bmod \text{١٠٥٥٥٧}$$
$$= \text{٤٩٥٨٣}$$
$$= m_{\text{٢}}.$$

- Using (٤.٢.٣.٣) method : If Ali wishes to send a message $M = (\text{٧٢٢٣٥}, \text{٤٩٥٨٣}) = (m_{\text{١}}, m_{\text{٢}})$ to Benin, then he should do the following:

–Compute $e\,Q = \text{٦٦٦١٢}(\text{٦٧١٥٣}, \text{١٠١١٧})$
$= (\text{٥٣١٣٤}, \text{٦٠٧٠٢})$
$= (k_{\text{١}}, k_{\text{٢}})$

–Compute $e\,B = \text{٦٦٦١٢}(\text{١٠٥٢٨٠}, \text{١٢٢٢٩})$
$= (\text{٨٦٣٢٨}, \text{١٥١٨٥})$

–Compute C:

$$C = (c_{\text{١}}, c_{\text{٢}})$$

$$c_{\text{١}} = (m_{\text{١}} + (k_1 k_2^{k_1})^{-1}) \bmod p$$
$$= (\text{٧٢٢٣٥} + (53134 * 60702^{53134})^{-1}) \bmod \text{١٠٥٥٥٧}$$
$$= (\text{٧٢٢٣٥} + (\text{٣١٢٢٣})^{-\text{١}}) \bmod \text{١٠٥٥٥٧}$$
$$= (\text{٧٢٢٣٥} + \text{٢٢٥٠٩}) \bmod \text{١٠٥٥٥٧}$$
$$= \text{٩٤٧٤٤}$$

$$c_{\text{٢}} = (m_{\text{٢}} + (k_{\text{٢}} k_{\text{١}}{}^{k}{}_{\text{٢}})^{-\text{١}} \quad \bmod p$$
$$= (\text{٤٩٥٨٣} + (53134 * 60702^{60702})^{-1}) \bmod \text{١٠٥٥٥٧}$$
$$= (\text{٤٩٥٨٣} + (\text{٥١٣٨٩})^{-\text{١}}) \bmod \text{١٠٥٥٥٧}$$
$$= (\text{٤٩٥٨٣} + \text{١٨٧٦٤}) \bmod \text{١٠٥٥٥٧}$$
$$= \text{٦٨٣٤٧}$$

Then Ali sends $C_m = \{C, e\,B\}$
$= \{(\text{٩٤٧٤٤}, \text{٦٨٣٤٧}), (\text{٨٦٣٢٨}, \text{١٥١٨٥})\}$ to Benin.

To decrypt the ciphertext, Benin should do the following:

–Compute $d\,(e\,B) = \text{٨٥٦١١}(\text{٨٦٣٢٨}, \text{١٥١٨٥})$
$= (\text{٥٣١٣٤}, \text{٦٠٧٠٢})$

$$= (k_{١},k_{٢})$$

–Compute M:

$$M = (m_{١}, m_{٢})$$

$$m_{١} = (c_{١} - (k_{١}k_{٢}^{k_{١}})^{-١}) \bmod p$$

$$= (٩٤٧٤٤ - ٢٢٥٠٩) \bmod ١٠٥٥٥٧$$

$$= ٧٢٢٣٥$$

$$= m_{١}$$

$$m_{٢} = (c_{٢} - (k_{٢}k_{١}^{k_{٢}})^{-١}) \bmod p$$

$$= (٦٨٣٤٧ - ١٨٧٦٤) \bmod ١٠٥٥٥٧$$

$$= ٤٩٥٨٣$$

$$= m_{٢}.$$

(٤. ٢. ٤) The Proposed Algorithms for Elliptic Curve Cryptosystems

In this subsection, we try to benefit from the DHEK for use this key (the key come from (DHEK algorithm) as a secret key in two suggestion methods.

As we look in *(٣.٥.١)*, Ali and Benin have the same point (only Ali and Benin know it). Then to start with Proposed Algorithm ١ (*PA*$_{١}$) and Proposed Algorithm ٢ (*PA*$_{٢}$) , let us consider the following algorithms:

Algorithm of (PA$_{١}$)

–Ali and Benin Compute $edB = S = (s_{١} , s_{٢})$.(using Diffie–Hellman Scheme)

–Ali sends a message $M \in E$ to Benin as follows:

– Compute $(s_{١} * s_{٢}) \bmod N = K.$ (such that $gcd(s_{١}* s_{٢},N)= ١$)[٢]

– Compute $K * M = C$, and send C to Benin.

–Benin receives C and decrypts it as follows:

– Compute $(s_{١} * s_{٢}) \bmod N = K.$

– Compute $(K^{-١}) \bmod N$. (where $N = \# E$)

– $K^{-١} C = K^{-١} * K * M = M.$

Algorithm of (PA$_{٢}$)

–Ali and Benin Compute $edB = S =(s_{١} , s_{٢})$.(using Diffie–Hellman Scheme)

[٢] If $gcd(s_{١}* s_{٢} ,N) \neq ١$, then Ali will search about a smallest integer number r such that $gcd((s_{١}* s_{٢}) + r,N)= ١$.

–Ali sends a message ***M*** to Benin as follows:

– Compute $(s_{١}^{s_{٢}}) \bmod N = K.$ (such that $gcd(s_{١}^{s_{٢}}, N)= ١$)[٢]

– Compute ***K * M*** = ***C***, and send ***C*** to Benin.

–Benin receives ***C*** and decrypts it as follows:

– Compute $(s_{١}^{s_{٢}}) \bmod N=K.$ (such that $gcd(s_{١}^{s_{٢}}, N)= ١$)

– Compute $(K^{-١}) \bmod N$.

– $K^{-١} C = K^{-١} * K * M = M.$

Example (٤.٥):

Let ***E*** be an EC defined over F_p where p = ٣٠٢٣ with parameters a= ١,b= ٢٥٤٧ where $(٤a^{٣}+ ٢٧b^{٢}) \bmod p$ = ٢٠٢٧ ≠ ٠. and # ***E*** = ٣٠٨٣, let ***B*** = ***(*** ٢٢٣٧, ٢٤٨٠***)*** is a base point on ***E.*** If Ali went to send a message ***M*** =***(*** ٢٢٨٤ , ٢٤٣٠ ***)*** to Benin using (*PA*١) and (*PA*٢) respectively , what did they do?

Solution

To apply this example using (*PA*١) , at first they must apply DHEK.

–Ali chooses a secret random integer ***e*** = ٢٣١٣.

eB = ٢٣١٣(٢٢٣٧, ٢٤٨٠) = (٩٣٤, ٢٩)

and sends ***(*** ٩٣٤, ٢٩***)*** to Benin .

–Benin chooses a secret random integer ***d*** = ١٢٣٦.

dB = ١٢٣٦(٢٢٣٧, ٢٤٨٠) = (١٧١٣, ١٧٠٩)

and sends ***(*** ١٧١٣, ١٧٠٩***)*** to Ali

–Ali computes the secret key ***e (dB)*** = ٢٣١٣(١٧١٣, ١٧٠٩).

edB = (٢٥٣٧, ١٦٣٢) = ***S***

–Benin computes the secret key ***d (eB)*** = ١٢٣٦(٩٣٤, ٢٩) .

deB = (٢٥٣٧, ١٦٣٢)= ***S***

Now, Ali and Benin have the same point ***S*** = ***(*** ٢٥٣٧, ١٦٣٢***).***

– Compute $(s_{١} * s_{٢}) \bmod N$ = (٢٥٣٧ * ١٦٣٢)***mod*** ٣٠٨٣

= ٢٩٩٨

= ***K.***

– Compute ***K* M*** = ٢٩٩٨(٢٢٨٤, ٢٤٣٠)

= (٢١٧٩, ١٨٣٢)

= ***C***, and send it to Benin.

–Benin receives ***C*** and decrypts it as follows:

– Compute $(s_{١} * s_{٢}) \bmod N$ = ٢٩٩٨ = ***K***

[٢] If $gcd(S_{١}^{S_{٢}} ,N) \neq ١$, then Ali will search about a smallest integer number r such that $gcd(S_{١}^{S_{٢}} + r,N)= ١$.

– Compute $(K^{-\text{١}})mod\ N = (\text{٢٩٩٨})^{-\text{١}}\ mod$ ٣٠٨٣
= ١٣٤٢

– $K^{-\text{١}}C$ = ١٣٤٢(٢١٧٩, ١٨٣٣)
= (٢٢٨٤, ٢٤٣٠).

To apply this example using the algorithm ($\mathcal{PA}_{\text{٢}}$), at first they must apply DHEK.

By the same procedure to solve Diffie – Helman scheme we have obtained S = (٢٥٣٧, ١٦٣٢). If Ali would like to send a message M = (٢٢٨٤ , ٢٤٣٠) to Benin using ($\mathcal{PA}_{\text{٢}}$) , then he should *do the following:*

– Compute $(s_1^{s_2})mod\ N = (\text{٢٥٣٧}^{\text{١٦٣٢}})\ mod$ ٣٠٨٣
= ٣٢٣ = K.

– Compute $K^* M$ = ٣٢٣(٢٢٨٤, ٢٤٣٠)
= (٢٥٥٥ , ١٠٦٦) = C, and send it to Benin.

–Benin receives C and decrypts it as follows:

–Compute $(s_1^{s_2})mod\ N$ = ٣٢٣ = K.

– Compute $(K^{-\text{١}})mod\ N = (\text{٣٢٣})^{-\text{١}}\ mod$ ٣٠٨٣
= ١٥٩٤.

– $K^{-\text{١}}C$ = ١٥٩٤(٢٥٥٥, ١٠٦٦)
= (٢٢٨٤, ٢٤٣٠)
= M.

Example (٤. ٦):

Let E be an EC defined over F_p where p = ١٠٥٥٥٧ with parameters a = ١١١١, b = ٢٢٢٤ where $(\text{٤}a^{\text{٣}} + \text{٢٧}b^{\text{٢}})mod\ p$ = ١٠٠٢١ ≠ ٠. and # E = ١٠٥١٤٣, let B = (١٠٤٧٦١, ٢١٣٣١) is a base point on E. If Ali wants to send a message M = (١٠٥٥٥١, ٨١٨٦٢) to Benin using ($\mathcal{PA}_{\text{١}}$) and ($\mathcal{PA}_{\text{٢}}$) respectively , what dose they do?

Solution

To apply this example using ($\mathcal{PA}_{\text{١}}$) , at first they must apply DHEK.

–Ali chooses a secret random integer e = ٩١٥٧٢.
eB = ٩١٥٧٢(١٠٤٧٦١, ٢١٣٣١) = (٤١٩٤, ٦٨٣٩٨)
and sends (٤١٩٤, ٦٨٣٩٨) to Benin .

–Benin chooses a secret random integer d = ٧٣٩٢٨.
dB = ٧٣٩٢٨(١٠٤٧٦١, ٢١٣٣١) = (٣٧٦٠٤, ٢٠٨٣٧)
and sends (٣٧٦٠٤,٢٠٨٣٧) to Ali

–Ali computes the secret key $e\,(dB)$ = ٩١٥٧٢(٣٧٦٠٤,٢٠٨٣٧).
edB = (٤٣٩١٨, ٤٠٦٠٠) = S

–Benin computes the secret key $d\,(eB)$ = ٧٣٩٢٨ (٤١٩٤, ٦٨٣٩٨) .

deB = (٤٣٩١٨, ٤٠٦٠٠) = S

Now, Ali and Benin have the same point S = (٤٣٩١٨, ٤٠٦٠٠).

= ($s_{١}$, $s_{٢}$)

– Compute ($s_{١}$ * $s_{٢}$) $mod\ N$ = (٤٣٩١٨ * ٤٠٦٠٠) mod ١٠٥١٤٣

= ٥٥٨٠٦

= K.

– Compute K* M = ٥٥٨٠٦ (١٠٥٥٥١, ٨١٨٦٢)

= (٢٧٨٢٥, ٨٧٧٢٤)

= C, and send it to Benin.

–Benin receives C and decrypts it as follows:

– Compute ($s_{١}$ * $s_{٢}$)$mod\ N$ = ٥٥٨٠٦ = K

– Compute ($K^{-١}$)$mod\ N$ = (٥٥٨٠٦)$^{-١}$ mod ١٠٥١٤٣

= ٧٤٩٢٨

– $K^{-١}$ C = ٧٤٩٢٨ (٢٧٨٢٥, ٨٧٧٢٤)

= (١٠٥٥٥١, ٨١٨٦٢).

= M.

To apply this example using the algorithm (*PA*٢), at first they must apply DHEK.

By the same procedure to solve Diffie – Helman scheme we have obtained S = (٤٣٩١٨, ٤٠٦٠٠). If Ali wishes to send a message M = (١٠٥٥٥١, ٨١٨٦٢) to Benin using (*PA*٢) , he should do the following:

– Compute ($s_1^{s_2}$)$mod\ N$ = (٤٣٩١٨ ^{٤٠٦٠٠}) mod ١٠٥١٤٣

= ٥٢٣٤٥ = K.

– Compute K* M = ٥٢٣٤٥ (١٠٥٥٥١, ٨١٨٦٢)

= (٥٣٣٦٣, ٧٤٢٦٦) = C,

and send C to Benin.

–Benin receives C and decrypts it as follows:

–Compute ($s_1^{s_2}$)$mod\ N$ = ٥٢٣٤٥ = K.

– Compute ($K^{-١}$)$mod\ N$ = (٥٢٣٤٥)$^{-١}$ mod ١٠٥١٤٣

= ٥٥٩٣٧.

– $K^{-١}$ C = ٥٥٩٣٧ (٥٣٣٦٣, ٧٤٢٦٦)

= (١٠٥٥٥١,٨١٨٦٢)

= M.

(٤.٣) Programming

In this subsection, we present the programming codes that are written to enhance the implementation of the above proposed methods and general treatment of ECCs calculator, where these software are written in MATLAB (The Language of Technical Computing) .

Program (١) Given E , prime p, $B=(b_{١},b_{٢})$, $Q=(q_{١},q_{٢})$, $M=(m_{١},m_{٢})$ and the secret key of Ali e, build the encryption scheme of ElGamal ECC algorithm.

```
clear
clc
Cm=[];
scmult(a,e,q١,q٢,p);
eq١=ans(١);
eq٢=ans(٢);
C=add(m١,m٢,eq١,eq٢,p);
scmult(a,e,b١,b٢,p);
eb١=ans(١);
eb٢=ans(٢);
eB=[eb١ eb٢];
Cm=[C], [eB]
```

Note ١. “a” is a coefficient of x in the equation of E.

٢. Ali sends “Cm” to Benin.

Program (٢) Given E , prime p, $B=(b_{١},b_{٢})$, $eB=(eb_{١},eb_{٢})$, $C=(c_{١},c_{٢})$ and the secret key of Benin d, build the decryption scheme of ElGamal ECC algorithm.

```
clear
clc
scmult(a,d,eb١,eb٢,p);
dq١=ans(١);
dq٢=ans(٢);
M=add(c١,c٢,dq١,-dq٢,p);
[M]
```

Note ١. “a” is a coefficient of x in the equation of E.

٢. “[M]” is the message.

Program (٣)

Given ***E*** , prime ***p***, ***B=(b١,b٢)***, ***Q=(q١,q٢)*** , ***M=(m١,m٢)*** and the secret key of Ali ***e***, build the encryption scheme of varying ElGamal ECC algorithm.

```
clear
clc
Cm=[];
scmult(a,e,q١,q٢,p);
eq١=ans(١);
eq٢=ans(٢);
C=add(m١,m٢,eq١,-eq٢,p);
scmult(a,e,b١,b٢,p);
eb١=ans(١);
eb٢=ans(٢);
eB=[eb١ eb٢];
Cm=[C], [eB]
```

Note ١. “a” is a coefficient of ***x*** in the equation of ***E***.
٢. Ali sends “Cm” to Benin.

Program (٤)

Given ***E*** , prime ***p***, ***B=(b١,b٢)***, ***eB=(eb١,eb٢)*** , ***C=(c١,c٢)*** and the secret key of Benin ***d***, build the decryption scheme of varying ElGamal ECC algorithm.

```
clear
clc
scmult(a,d,eb١,eb٢,p);
dq١=ans(١);
dq٢=ans(٢);
M=add(c١,c٢,dq١,dq٢,p);
[M]
```

Note ١. “a” is a coefficient of ***x*** in the equation of ***E***.
٢. “[M]” is the message.

Program (٥)

Given ***E*** , prime ***p***, ***B=(b*** ***١******,b*** ***٢******)***, ***Q=(q*** ***١******,q*** ***٢******)*** , ***M=(m*** ***١******,m*** ***٢******)*** and the secret key of Ali ***e***, build the encryption scheme of MVECC algorithm.

```
clear
clc
scmult(a,e,q١,q٢,p);
k١=ans(١);
k٢=ans(٢);
c١=mod(m١*k١,p);
c٢=mod(m٢*k٢,p);
C=[c١ c٢];
scmult(a,e,b١,b٢,p);
eb١=ans(١);
eb٢=ans(٢);
eB=[eb١ eb٢];
Cm=[C],[eB]
```

Note ١. “a” is a coefficient of ***x*** in the equation of ***E***.

٢. Ali sends “Cm” to Benin.

Program (٦)

Given ***E*** , prime ***p***, ***B=(b*** ***١******,b*** ***٢******)***, ***eB=(eb*** ***١******,eb*** ***٢******)*** , ***C=(c*** ***١******,c*** ***٢******)*** and the secret key of Benin ***d***, build the decryption scheme of MVECC algorithm.

```
clear
clc
scmult(a,d,eb١,eb٢,p);
k١=ans(١);
k٢=ans(٢);
m١=mod(c١*invmod(k١,p),p);
m٢=mod(c٢*invmod(k٢,p),p);
M=[m١ m٢];
[M]
```

Note ١. “a” is a coefficient of ***x*** in the equation of ***E***.

٢. “[M]” is the message.

Program (٧)

Given ***E*** , prime ***p***, $B=(b_١,b_٢)$, $Q=(q_١,q_٢)$, $M=(m_١,m_٢)$ and the secret key of Ali ***e***, build the encryption scheme of proposition MVECC١ algorithm.

```
clear
clc
scmult(a,e,q١,q٢,p);
k١=ans(١);
k٢=ans(٢);
c١=mod(m١ + k١*k٢,p);
c٢=mod(m١*(m٢ + k١*k٢),p);
C=[c١ c٢];
scmult(a,e,b١,b٢,p);
eb١=ans(١);
eb٢=ans(٢);
eB=[eb١ eb٢];
Cm=[C],[eB]
```

Note ١. “a” is a coefficient of ***x*** in the equation of ***E***.
٢. Ali sends “Cm” to Benin.

Program (٨)

Given ***E*** , prime ***p***, $B=(b_١,b_٢)$, $eB=(eb_١,eb_٢)$, $C=(c_١,c_٢)$ and the secret key of Benin ***d***, build the decryption scheme of proposition MVECC١ algorithm.

```
clear
clc
scmult(a,d,eb١,eb٢,p);
k١=ans(١);
k٢=ans(٢);
m١=mod(c١- k١*k٢,p);
m٢=mod(invmod(m١,p)*c٢-k١*k٢,p);
M=[m١ m٢];
[M]
```

Note ١. “a” is a coefficient of ***x*** in the equation of ***E***.
٢. “[M]” is the message.

Program (٩)

Given ***E*** , prime ***p***, $\boldsymbol{B=(b_{١},b_{٢})}$, $\boldsymbol{Q=(q_{١},q_{٢})}$, $\boldsymbol{M=(m_{١},m_{٢})}$ and the secret key of Ali ***e***, build the encryption scheme of proposition MVECC٢ algorithm.

```
clear
clc
scmult(a,e,q١,q٢,p);
k١=ans(١);
k٢=ans(٢);
c١=mod(m١ *(k١*k٢ - k١),p);
c٢=mod(m٢ *(k١*k٢ - k٢),p);
C=[c١ c٢];
scmult(a,e,b١,b٢,p);
eb١=ans(١);
eb٢=ans(٢);
eB=[eb١ eb٢];
Cm=[C],[eB]
```

Note ١. “a” is a coefficient of ***x*** in the equation of ***E***.

٢. Ali sends “Cm” to Benin.

Program (١٠)

Given ***E*** , prime ***p***, $\boldsymbol{B=(b_{١},b_{٢})}$, $\boldsymbol{eB=(eb_{١},eb_{٢})}$, $\boldsymbol{C=(c_{١},c_{٢})}$ and the secret key of Benin ***d***, build the decryption scheme of proposition MVECC٢ algorithm.

```
clear
clc
scmult(a,d,eb١,eb٢,p);
k١=ans(١);
k٢=ans(٢);
m١=mod(c١ * invmod(k١*k٢ - k١,p),p);
m٢=mod(c٢ * invmod(k١*k٢ - k٢,p),p);
M=[m١ m٢];
[M]
```

Note ١. “a” is a coefficient of ***x*** in the equation of ***E***.

٢. “[M]” is the message.

Program (١١) Given ***E*** , prime ***p***, ***B=(b١,b٢)***, ***Q=(q١,q٢)*** , ***M=(m١,m٢)*** and the secret key of Ali ***e***, build the encryption scheme of proposition MVECC٣ algorithm.

```
clear
clc
scmult(a,e,q١,q٢,p);
k١=ans(١);
k٢=ans(٢);
R١=mod(fastexp(k١*k٢,k١,p),p);
R٢=mod(fastexp(k١*k٢,k٢,p),p);
c١=mod(m١ + invmod(R١,p),p);
c٢=mod(m٢ + invmod(R٢,p),p);
C=[c١ c٢];
scmult(a,e,b١,b٢,p);
eb١=ans(١);
eb٢=ans(٢);
eB=[eb١ eb٢];
Cm=[C],[eB]
```

Note ١. “a” is a coefficient of ***x*** in the equation of ***E***.

٢. Ali sends “Cm” to Benin.

Program (١٢) Given ***E*** , prime ***p***, ***B=(b١,b٢)***, ***eB=(eb١,eb٢)*** , ***C=(c١,c٢)*** and the secret key of Benin ***d***, build the decryption scheme of proposition MVECC٣ algorithm.

```
clear
clc
scmult(a,d,eb١,eb٢,p);
k١=ans(١);
k٢=ans(٢);
R١=mod(fastexp(k١*k٢,k١,p),p);
R٢=mod(fastexp(k١*k٢,k٢,p),p);
m١=mod(c١ - invmod(R١,p),p);
m٢=mod(c٢ - invmod(R٢,p),p);
M=[m١ m٢];
[M]
```

Note ١. “a” is a coefficient of ***x*** in the equation of ***E***.

٢. “[M]” is the message.

Program (١٣)

Given E , prime p, $S=(s_{١},s_{٢})$, $M=(m_{١},m_{٢})$ and $\#E$, build the encryption scheme of PA$_{١}$.

```
clear
clc
for r=٠:N
    if gcd((s١ * s٢)+r,N)==١
        K=mod((s١ * s٢)+r,N);
        C=scmult(a,K,m١,m٢,p);
        [C], break
    else r=r+١;
    end
end
```

Note ١. “N” = ***#E.***

٢**.** “a” is a coefficient of x in the equation of E.

٣. Ali sends “C” to Benin.

Program (١٤)

Given E , prime p, $S=(s_{١},s_{٢})$, $C=(c_{١},c_{٢})$ and $\#E$, build the decryption scheme of PA$_{١}$.

```
clear
clc
for r=٠:N
    if gcd((s١ * s٢)+r,N)==١
        K=mod((s١ * s٢)+r,N);
        inv(K)=invmod(K,N);
        M=scmult(a,inv(K),c١,c٢,p);
        [M], break
    else r=r+١;
    end
end
```

Note ١. “N” = ***#E.***

٢**.** “a” is a coefficient of x in the equation of E.

٣. Ali sends “C” to Benin.

Program (١٥)

Given ***E*** , prime ***p***, ***$S=(s_{١},s_{٢})$***, ***$M=(m_{١},m_{٢})$*** and ***#E***, build the encryption scheme of $PA_{٢}$.

```
clear
clc
s١٢=fastexp(s١,s٢,N);
for r=٠:N
      if gcd(s١٢+r,N)==١
        K=mod((s١٢)+r,N);
        C=scmult(a,K,m١,m٢,p);
        [C], break
    else r=r+١;
    end
end
```

Note ١. “N” = ***#E.***

٢**.** “a” is a coefficient of ***x*** in the equation of ***E***.

٣. Ali sends “C” to Benin.

Program (١٦)

Given ***E*** , prime ***p***, ***$S=(s_{١},s_{٢})$*** , ***$C=(c_{١},c_{٢})$*** and ***#E***, build the decryption scheme of $PA_{٢}$.

```
clear
clc
s١٢=fastexp(s١,s٢,N);
for r=٠:N
    if gcd(s١٢+r,N)==١
        K=mod(s١٢+r,N);
        inv(K)=invmod(K,N);
        M=scmult(a,inv(K),c١,c٢,p);
        [M], break
    else r=r+١;
    end
end
```

Note ١. “N” = ***#E.***

٢**.** “a” is a coefficient of ***x*** in the equation of ***E***.

٣. Ali sends “C” to Benin.

Program (١٧)

Given any natural number ***n***, compute all quadratic residues modulo ***n***.

```
clc
clear
for b=١:n-١
   for c=١:n-١
      if mod(b,n)==mod(c^٢,n) [b  c], break
      end
   end
end
```

Note ١. "[b c]" means print the values of b and c.

٢. "b" denotes the quadratic residues modulo ***n***.

Program (١٨)

Given any positive integer ***n***, compute order of ***a mod n***, for all ***a*** = ***١:n- ١***

```
clear
clc
for a=١:n-١
   for i=١:n-١
      z=a^i;
      if mod(z,n)==١  [a  i] , break
      end
   end
end
```

Note ١. "i" denotes the order of ***a mod n***, for all ***a*** = ***١:n- ١.***

Program (١٩)

Given any positive integer ***n***, compute order of ***a mod n .***

```
clc
clear
for k=١:n-١
   if mod(a^k,n)==١ [k] ,break
   end
end
```

Note ١. "k" denotes the order of ***a mod n***,

Program (٢٠) Given any positive integer n, compute Euler's fun. for n .

```
clear
clc
c=٠;
for k=١:n -١
   if gcd(n,k)==١   , c=c+١;
   end
end
[c]
```

Note ١. “c” denotes to $\Phi(n)$.

Program (٢١) Given a prime p, compute all Elliptic Curves over F_p and the number of it.

```
clc
c=٠;
for a=٠:p
    for b=٠:p
        if mod(٤*a^٣+٢٧*b^٢,p)~=٠, c=c+١;  [a, b]
        end
    end
end
[c]
```

Note ١. “a, b” denotes a , b in (٩) respectively.

٢. “c” denotes number of the pointes on E .

Program (٢٢)

Given ***E*** and prime ***p,*** compute all point ***(x,y)*** on ***E*** and the number of it.

```
clear
clc
c=١;
for x=٠:p-١
    for y=٠:p-١
    z=y.^٢;
    w=x.^٣+a*x+b;
    if mod(z,p)== mod(w,p) ,c=c+١; [x,y]
    end
end
end
[c]
```

Note ١. “[x,y]” denotes the pointes on ***E***.

٢. “c” denotes number of it .

Program (٢٣)

Given ***E***, prime ***p***, and x-coordinate compute the y-coordinate.

```
clear
clc
for y=٠:p-١
    if mod(y^٢,p)==mod(fastexp(x,٣,p)+a*x+b,p),[x ,y]
    end
end
```

Program (٢٤)

Given ***E***, prime ***p***, $Q(x_{٢},y_{٢})$,and $P(x_{١},y_{١})$, compute all ***k*** where $Q = kP$.

```
clear
clc
for k=٠:p-١
    z=[];
    oe=sym('OE');
    r=dec٢bin(k);
    [row,col]=size(r);
    xk=x١;
    yk=y١;
    for i=٢:col
        doup(xk,yk,a,p);
        xk=ans(١);
        yk=ans(٢);
        if r(i)==٤٩
            add(xk,yk,x١,y١,p);
            xk=ans(١);
            yk=ans(٢);
        end
    end
    if xk==x٢&yk==y٢ , [k]
    end
end
```

Program (٢٥)

Given ***E***, prime ***p***, $P(x_{١},y_{١})$,and ***#(E)*** compute ***k*P*** ,for more ***k***.

```
clear
clc
z=[];
oe=sym('OE');
for k=٠:N
    r=dec٢bin(k);
    [row,col]=size(r);
    xk=x١;
    yk=y١;
    for i=٢:col
        doup(xk,yk,a,p);
        xk=ans(١);
        yk=ans(٢);
        if r(i)==٤٩
            add(xk,yk,x١,y١,p);
            xk=ans(١);
            yk=ans(٢);
        end
    end
    z=[z;xk yk k];
end
[z]
```

Program (٢٦)

Given ***E***, prime ***p***, ***P(x*****١*****,y*****١*****),*** compute the order of ***P.***

```
clear
clc
oe=sym('OE');
for k=١:٢*p
    r=dec٢bin(k);
    [row,col]=size(r);
    xk=x١;
    yk=y١;
    for i=٢:col
        doup(xk,yk,a,p);
        xk=ans(١);
        yk=ans(٢);
        if r(i)==٤٩
            add(xk,yk,x١,y١,p);
            xk=ans(١);
            yk=ans(٢);
        end
    end
    if xk==oe &yk==oe [k], break
    end
end
```

Note ١. “[k]” denotes the order of ***P***.

Chapter Five

Discussion and Conclusions

(٥. ١) Discussion

This section concludes the brief review of the chapters two, three and four. Let us organize these briefs as follows:

In chapter two, we explain some of algebraic preliminaries and some of the basic concepts of the number theory which are needed for researcher in ECCg. and also define the equation of an EC over a field and explain all the operations laws of the group of points on ECs over three fields.

In chapter three, we describe the two different classes of cryptosystem, private and public key cryptosystem, and the details of the following ECCs with examples:

- ***Elliptic Curve Diffie – Hellman Key Exchange,***
- ***Elliptic Curve Massey – Omura cryptosystem (ECMO),***
- ***ElGamal Elliptic Curve Cryptosystem,***
- ***Menezes – Vanstone Elliptic Curve Cryptosystem (MVECC),***
- ***Elliptic Curve Digital Signature Algorithm (ECDSA).***

This chapter also contains the famous attack methods with some examples. It is arranged as follows:

- ***Exhaustive Search,***
- ***Parallel Pollard rho Method,***
- ***Pohlig-Hellman Attack.***

Chapter four, −which we esteem as implementation chapter in the thesis− contains the following items:

- Variant ElGamal ECCs with same complexity,
- Development the encryption and decryption MVELC scheme in four propositions,
- Suggestion of two methods to encrypt and decrypt messages using the properties of ECs,
- Design of program to enhance the implementation of the above proposed methods and general treatment of ECCs calculator, where these software are written in MATLAB (The Language of Technical Computing) , version ٦.٥ on computer type P٤ with ١.٧ GHz.

(٥. ٢) Conclusions

((١)) The details of EC are very easy when we compute them by some programs. We make some programs to compute the following concepts:

- The quadratic residues of any prime number ,
- The order of ***a* mod *n***, for all a = ١ to n- ١,
- The order of ***a* mod *n*** for any integer number ***a,***
- The $\Phi(n)$ for all ***n,***
- All ***a,b*** such that $٤a^{٣}+ ٢٧b^{٢} \bmod p =$ ٠,
- All EC over F_p and the number of it,
- All point (x,y) on the EC, and number of it,
- The y–coordinate when the x–coordinate is given,
- Find all k such that $Q = kP$ for all k = ١ to p- ١,
- The $k*P$, for more k.
- The order of any point on EC,

((٢)) When we study The scheme of ElGamal ECCs, we see that this scheme cannot be development, but we can vary it with same complexity, as in the proposition (٤. ٢. ٢) , because the addition and subtraction have the same computation complexity.

((٣)) From the advantages of the MVECC scheme[١] , then we can use this to the encryption and decryption scheme, as in the

[١] ♦It is not depend on addition operation on elliptic curve group,
♦The message needs not to be a point on elliptic curve.

propositions *(٤.٢.٣.١)*, *(٤.٢.٣.٢)*, and *(٤.٢.٣.٣)*, with the complexity given as follows:

- The proposition *(٤.٢.٣.١)* is more efficient than the ***MVECC*** and the proposed method ١ in [٤٣] , where in the encryption scheme we do three multiplication operations $(k_{١} k_{٢}, m_{١} m_{٢}$ and $m_{١} k_{٢} k_{١})$ and two addition operations. And the decryption scheme needed to compute the inverse operations for $m_{١}$, and three multiplication operations $(k_{١} k_{٢}, m_1^{-1} c_{٢}$ and $m_1^{-1} k_{١} k_{٢})$ and two subtraction operations.
- The proposition *(٤.٢.٣.٢)* is more efficient than the previous proposition *(٤.٢.٣.١)* and than the proposed method ٢ in [٤٣] , where in the encryption scheme we do three multiplication operations $(k_{١} k_{٢}, m_{١} k_{٢} k_{١}$ and $m_{٢} k_{٢} k_{١})$ and two subtraction operations. And the decryption scheme needed to compute the inverse operations for $(k_{١} k_{٢} - k_{١}$ and $k_{١} k_{٢} - k_{٢})$ and two multiplication operations $(c_{١} * (k_{١} k_{٢} - k_{١})^{-١}$ and $c_{٢} * (k_{١} k_{٢} - k_{٢})^{-١})$.
- The proposition *(٤.٢.٣.٣)* is more efficient than the previous propositions *(٤.٢.٣.١)* and *(٤.٢.٣.٢)* and than the last two proposed methods in [٤٣] , where in the encryption scheme we use the exponentiation operation between two keys to make the key more secure ($k_2^{k_1}$ and $k_1^{k_2}$), and this scheme needed to compute the inverse operations for $(k_{١}\ k_2^{k_1}$ and $k_{٢}\ k_1^{k_2})$ also two addition operations. the decryption scheme needed to compute all the above operations in the encryption scheme and and two subtraction operations.

((٤)) We benefit from the ***Diffie–Hellman Exchanging key*** for use this key (the key come from (Diffie–Hellman Exchanging key algorithm) as a secret key in two suggestion methods.

(٥.٣) Future Works

((١)) The possibility of applying the hyperelliptic curve of different genus over a finite field to implement an ECCss.

((٢)) Using the index calculus algorithm for Discrete Logarithm Problem on EC . this construction will be based on the Problem finding bounded solutions to some explicit modular multivariate polynomial equations.

((٣)) Determination the number of isomorphism classes of Picard Curve, i.e. superelliptic curves $y^{٣} = f(x)$ of genus more than ٢ over finite field.

References

[١] Burton, D.M. (١٩٧٦), “***Elementary Number Theory***”, Second Edition, University of New Hampshire.

[٢] Whitfield, D. and Martin, H. (١٩٧٦), “ ***New Directions in Cryptography***”.

[٣] Meng, T.K. (٢٠٠١),“ ***Curves For The Elliptic Curve Cryptosystem***”, M.S.C. Thesis, University of Singapore.

[٤] http://www.certiom.com/index.php?action=ecctutorial,home

[٥] Wiener, M.J. and Zuccherato, R. J. (١٩٩٨),“ ***Faster Attacks on Elliptic Curve Cryptosystems***”, Entrust Technologies, Canada.

[٦] Hankerson, D. and Menezes, A. (٢٠٠٣), “ ***Elliptic Curve Cryptography***”, University of Waterloo.

[٧] Saeki, M.K. (١٩٩٧),“ ***Elliptic Curve Cryptosystems***”, M.S.C. Thesis, McGill University, Montreal.

[٨] Robshaw, M.J.B. and Yin, Y.L. (١٩٩٧),“ ***Elliptic Curve Cryptosystems***”, An RSA Laboratories Technical Note.

[٩] Sutikno, S. , Surya, A. and Effendi, R. (١٩٩٨),“ ***An Implemantation of ElGamal Elliptic Curve Cryptosystems***”, Integrated System Labaratory, pp. ٤٨٣–٤٨٦.

[١٠]Hills, J. (٢٠٠١),),“ ***Introduction to Elliptic Curve Cryptography***”, A Mathematical Overview.

[١١]Reis, J.V (٢٠٠١), “***Elliptic Curves***”.

[١٢]Reis, J.V (٢٠٠١), “***Elliptic Curves over Fields of Characteristic ٣***”.

[١٣]Oswald, E. (٢٠٠٢), “***Introduction to Elliptic Curve Cryptography***”, Institute for Applied Information Processing and Communication, Graz University Technology.

[١٤]Packard, M.R. (٢٠٠٢), “***Elliptic Curve Cryptography***”.

[١٥]Chouinard, J.Y. (٢٠٠٢), “***Elliptic Curve Cryptography***”, Design of Secure Computer System.

[١٦]Evans, P.D. (٢٠٠١), “***Timing attack on Elliptic Curve Cryptography***”, University of Virginia.

[١٧]Seroussi, G. (١٩٩٨), “***Compact Representation of Elliptic Curve Points over*** $F_{٢}^{n}$ ”, Computer Systems Laboratory, Hewlett-Packard Company.

[١٨]Sutikno, S. and Surya, A. (٢٠٠٠), “***An Architecture of*** $F_{٢}^{٢n}$ ***Multiplier for Elliptic Curve Cryptosystem***”, IEEE International Symposium on Circuits and Systems, Geneva, Switzerland.

[١٩]Gathani, A.N. (٢٠٠١), “***Implemantation of Elliptic Curve Cryptography In Imbedded System, A Brief Overview***”.

[٢٠]Weimerskirch, A., Stebila, D. and Shantz, S.C. (٢٠٠٣), ***“Generic GF(٢m) Arithmetic in Software and its Application to ECC”***, Wollongong, Australia.

[٢١]Nguyen, N., Gaj, K., Caliga, D. and El-Ghazawi, T. (٢٠٠٣) , “***Implementation of Elliptic Curve Cryptosystems on a Reconfigurable Computer***”, Second IEEE International Conference on Field–Programmable Technology, Japan.

[٢٢]Murari, A. (٢٠٠٣), “***Software Implementations of Elliptic Curve Cryptography***”, Oregon State university.

[٢٣]Arslanian, S.T. (١٩٩٨), “ ***An Implemantation of ElGamal Elliptic Curve Cryptosystems over a Finite Field of Characteristic p***”, M.S.C. Thesis, University of Maine.

[٢٤]Fibỉýkovả, L. (٢٠٠٢), “***Elliptic Curve Cryptography Over Prime Fields***”, University of Essen, Germany.

[٢٥]Hankerson, D. and Menezes, A. (٢٠٠٣), “***Elliptic Curve Public–Key Encryption Schemes***”, University of Waterloo.

[٢٦]IAIK's ECC Library (٢٠٠٣), "***Elliptic Curve Cryptography***", Graz, University of Technology.

[٢٧]Huang, Q., Kobayashi, H., Liu, B., Gu, D. and Zhang, J. (٢٠٠٤), "***Energy/Security Scalable Mobile Cryptosystem***"http://www.merl.com

[٢٨]Certicom Corp., (١٩٩٧), "***Current Public–Key Cryptographic SystemsAn Introduction to Information Security***", Certicom White Papers, number ٢.

[٢٩]Certicom Research (١٩٩٩), "***Elliptic Curve Cryptography***", Standards For Efficient Cryptography, Version ٠.٥.

[٣٠]Pietiläinen, H. (٢٠٠٠), "***Elliptic Curve Cryptography on smart cards***", M.S.C. Thesis, University of Technology.

[٣١]Torii, N. and Yokoyama, K. (٢٠٠٠), "***Elliptic Curve Cryptosystem***", FUJITSU Sci. Tech. J., pp.١٤٠-١٤٦.

[٣٢]Menezes, A., Okomoto, T. and Vanstone, S.A. (١٩٩٣), "***Reduucing Elliptic Curve Logarithms to Logarithms in a Finite Field***", IEEE Transacton on Information Theory, vol. ٣٩, no. ٥, pp.١٦٣٩–١٦٤٦.

[٣٣]Certicom Corp., (١٩٩٧), "***Remarks on the Security of the Elliptic Curve Cryptosystem*** ", Certicom White Papers, number ٥.

[٣٤]Griffeth, S. (٢٠٠١), "***Why Count Points?*** ".

[٣٥]Menezes, A. (٢٠٠١), "***Evaluation of Security Level of Cryptography:The Elliptic Curve Discrete Logarithm Problem*** ", University of Waterloo.

[٣٦]York, E.V. (١٩٩٢), "***Elliptic Curves Over Finite Fields***", George Mason University.

[٣٧]Griffeth, S. (٢٠٠١), "***How to Count Points on Elliptic Curves, part*** I ***and*** II".

[٣٨]Ros, M.B. (٢٠٠٠), "***Hyperelliptic Curve Cryptography over Optimal Extension Fields***", M.S.C. Thesis, University of Queensland.

[٣٩]Wollinger, T. (٢٠٠١), “***Computer Architectures for Cryptosystems Based on Hyperelliptic Curves***”, M.S.C. Thesis, Worcester Polytechnic Institute.

[٤٠]McLeman, C. (٢٠٠١), “***Hyperelliptic Curves***”.

[٤١]Donis, P. (٢٠٠١), “***Choosing Elliptic Curves Suitable For Cryptographic Applications***”, George Mason University.

[٤٢] العزاوي، شاكر محمود (٢٠٠٣)، ”تقويم فعالية المنحنيات الاهليليجية في انظمة التشفير“ ، اطروحة ماجستير، الجامعة التكنولوجية، بغداد.

[٤٣]Sagheer, A.M. (٢٠٠٤), “***Enhancement of Elliptic Curves Cryptography Methods***”, M.S.C. Thesis, University of Technology, Baghdad.

[٤٤]Radi, S.K. (٢٠٠١), “***On Primality Tasting***”, M.S.C. Thesis, University of Technology, Baghdad.

[٤٥]Nasir, Z.T. (٢٠٠٠), “***Primality Tasting Using Elliptic Curve***”, M.S.C. Thesis, University of Technology, Baghdad.

[٤٦]Yan, S.Y. (٢٠٠٠), “***Number Theory For Computing***”, Springer, Berlin.

[٤٧] بيرتون، ديفد م. (١٩٦٧)، ”مقدمة في الجبر المجرد الحديث“ ، جامعة هامشاير.

[٤٨]“***Polynamial Representation***”, http://www.certiom.com/index.php?action=ecctutorial,home

[٤٩]“***Optimal Normal Basis***”, http://www.certiom.com/index.php?action=ecctutorial,home

[٥٠]Rosen, K.H. (٢٠٠٠), “***Elementary Number Theory and its Applications***”, Fourth Addition, Addition Wesley Longman.

[٥١]Milne, J.S. (١٩٩٦), “***Elliptic Curves***”, vol. ١.٠١.

[٥٢]Finney, R.L. and Thomas, G.B. (١٩٩٠), “***Calculus***”, vol. ١, U.S.A.

[٥٣]Silverman, J.H. (١٩٨٦), “***The Arithmetic of Elliptic Curves***”, Springer–Verlag, New York.

[٥٤]Koeller, J., Menezes, A., Qu, M. and Vanstone, S.A. (١٩٩٦), ***Elliptic Curves Systems*** ”, Canada.

[٥٥]Monnerat, J. (٢٠٠٢), “***Computation of the Discrete Logarithm on Elliptic Curves of Trace One***”, Security and Cryptography Laboratory, Switzerland.

[٥٦]Rolland, R. (٢٠٠١), “***Public key cryptography***”.

[٥٧]Zhu, H. (٢٠٠١)), “***Survey of Computational Assumptions Used in Cryptography Broken or Not by Shor's Algorithm***”, M.S.C. Thesis, McGill University, Canada.

[٥٨]“***Menezes–Vanstone Elliptic Curve Cryptosystem***”, http://www–fs.informatik.uni–tuebingen/~reinhard/krpto/java/ Menezes–Vanstone Elliptic Curve Cryptosystem–e.java.

[٥٩]Jurisic, A. and Menezes, A. , “***Elliptic Curves and Cryptography***”.

[٦٠]Aydos,M., Savaş, E. and Koç, Ç.K. (١٩٩٩), “***Implementing Network Security Protocols based on Elliptic Curve Cryptography***”, Oregon State University, U.S.A.

[٦١]Studholme, C. (٢٠٠٢), , “***The Discrete Log Problem***”.

[٦٢]Hankerson, D. and Menezes, A. (٢٠٠٣) “***Elliptic Curve Discrete Logarithm Problem***”, University of Waterloo.

[٦٣]Certicom Corp.(١٩٩٨), “The Elliptic Curve Cryptosystem for Smart Cards”, Certicom White Papers, number ٧.