ECC: Elliptic Curve Cryptography

Alvarez G. A.
arturo12gaona@gmail.com

May 18, 2021

Introduction

The following document describes the basic theory about elliptic curves for solving the Capture The Flag (CTF) practice of the ECC System. Likewise, it is assumed that the reader has the algebraic bases and finite field theory (otherwise you can consult [1] [2]) in order to understand how elliptic curves can be applied to cryptography. The document describes the definition of what is an elliptic curve on real numbers through examples and simple graphs, then we move on to the description on the elliptic curves in binary finite fields or characteristic 2, continuing with the operations on binary finite fields in elliptic curves through the equations defined by Weierstrass. Finally, simple examples of the applications of elliptic curves for the field of public key cryptography and their use in digital signatures are shown.

Definition of elliptic curves

Elliptic curves are defined by cubic equations. These have been studied for several years, they are currently used in cryptography. When working with elliptic curves, a binary operation can be defined on the set of points in a geometric way, which makes said set an abelian group.

Definition An elliptic curve is of the form [3]:

$$E : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_5$$

Where a_1,..., a_5 are constant elements that belong to a field.

Elliptic curves over the real number field $\mathbb{R}$

An elliptic curve over $\mathbb{R}$, is defined as the set of points (x, y) with $x, y \in \mathbb{R}$ that satisfy the simplified Weierstrass equation $E : y^2 = x^3 + ax + b$, where $a,\ b\ \in \mathbb{R}$, and $\Delta = 4a^3 + 27b^2$ is the discriminant [4]. If $\Delta \neq 0$, the curve has no repeating roots. For each pair of values (a, b) there is a different elliptic curve.

For example, if we take the value of $a = -11$ and $b = 4$ the geometric representation is the one shown in Figure 1.

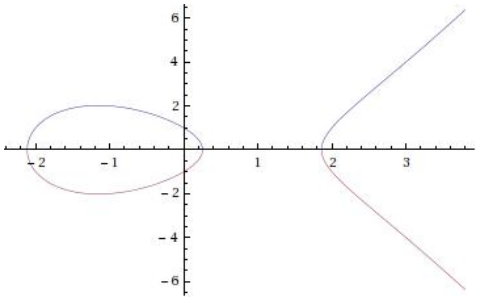

Figure 1: $y^2 = x^3 - 11x + 4$

Now if we take the value of $a = -3$ y $b = 2$, the discriminant $\Delta = 0$ and its geometric representation can be seen in Figure 2.

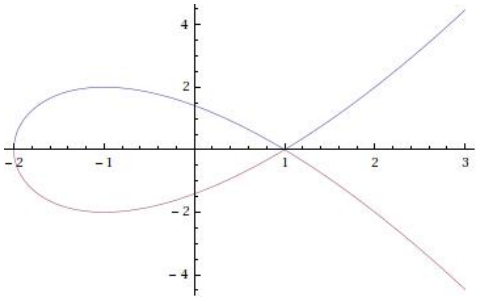

Figure 2: $y^2 = x^3 - 3x + 2$ with $\Delta = 0$

Sum of points on an elliptic curve E over the field $\mathbb{R}$

Let two points P and Q lie on the curve E, a straight line as drawn through these two points, the result will be a new intersection on the curve, this new point will be $-R$. Now if we draw a vertical line over the point $-R$ the result is another point R, for the general case we can define the sum of two points as $R = P + Q$. Figure 3.

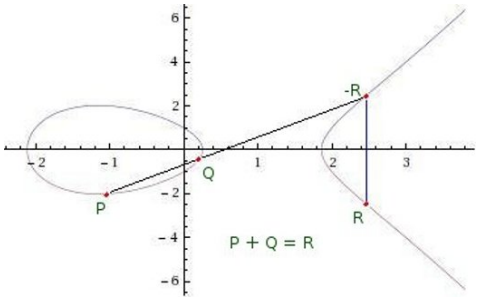

Figure 3: Sum of points on the curve $y^2 = x^3 - 11x + 4$

Inverse element in an elliptic curve over the field $\mathbb{R}$

Figure 3 also shows the negative of point R. The point $R = (x, y)$ is geometrically the reflection of point y, that is; $-R = (x, -y)$.

Identity element on an elliptic curve over $\mathbb{R}$

Consider a point P on the curve and if we want to know with which another point added gives me the same point, we need an extra point called *infinity point* and we will denote it by ϑ.
It can say that the infinity point ϑ is infinitely far on the vertical axis and is the *identity* in an elliptic curve, and fulfilling the property $P + \vartheta = P$.

Likewise we have that given $P = (x, y)$ and its inverse $-P = (x, -y)$ the sum of the points results in the point at infinity $P + (-P) = \vartheta$. Figure 4.

Product of an integer n with a point on an elliptic curve over $\mathbb{R}$

The product between a point $P(x, y) \in E$ with $y \neq 0$ and a integer n consists of adding the point P, n times. When $n = 1$ we have that: $1P = P$. If $n = 2$ then a tangent to the curve is drawn at point P. The tangent intersects the curve at a second point $-R$ and therefore $2P = R$, Figure 5. For $n = 3$ is calculated $3P = 2P + P$.

If $y = 0$, the tangent is vertical and does not intersect the curve at more points. By definition $2 * P = \vartheta$, Figure 6. In general $n * P = \vartheta$ if n is even and $n * P = P$ if n is odd when $y = 0$.

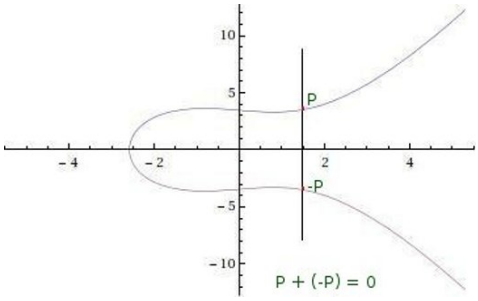

Figure 4: Sum of point P with its inverse $-P$ over the curve $y^2 = x^3 - 2x + 12$

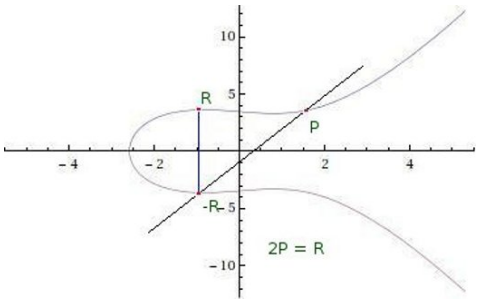

Figure 5: Sum of the point P with itself of the curve $y^2 = x^3 - 2x + 12$

The $\mathbb{R}$ field is not considered practical for cryptography due to errors that can occur in rounding values. Instead, finite prime fields $\mathbb{F}_p$ (or also with the notation $GF(p)$ by Galois field) are used where p is a prime number and finite fields of the form $\mathbb{F}_{2^m}$ (or also with the notation $GF(2^m)$ by Galois field), $m \geq 1$, which are a very reliable alternative in modern cryptography [5]. For this practice, we will consider the fields of the form $\mathbb{F}_{2^m}$.

Definition of elliptic curves on $\mathbb{F}_{2^m}$

The finite fields $\mathbb{F}_{2^m}$ with $m \geq 1$ are very convenient, since being characteristic 2, the coefficients of the polynomials can be represented as a string of bits, facilitating their implementation in hardware.

Law $E(k)$

An elliptic curve can be classified according to the characteristics of its base K and its discriminant Δ. For curves defined on $\mathbb{F}_{2^m}$ they are classified as [6]:

- Non-supersingular
  - The curve is defined as: $y^2 + xy = x^3 + ax^2 + b$
  - $\Delta = b$
- Supersingular
  - The curve is defined as: $y^2 + cy = x^3 + ax + b$
  - $\Delta = c^4$

Where a, b and $c \in \mathbb{F}_{2^m}$.

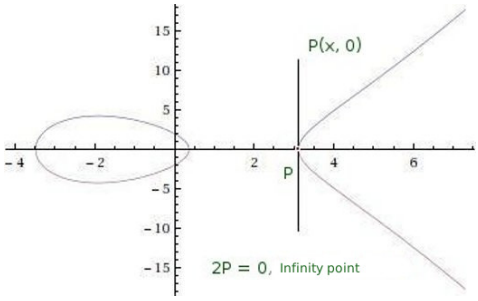

Figure 6: Sum of the point P with itself with the coordinate $y = 0$ of the curve $y^2 = x^3 + 4x + -7$

Elliptic curves on $\mathbb{F}_{2^m}$

We will work with non-supersingular curves and with the Weierstrass equation over $\mathbb{F}_{2^m}$ [7], conforms to:

$$E : y^2 + xy = x^3 + ax^2 + b \tag{1}$$

Where $a, b \in \mathbb{F}_{2^m}$ with $\Delta = b \neq 0$. A point $P(x,y)$ lies on E if it satisfies the equation.

Example:

To know which points belong to the curve, the finite field $\mathbb{F}_{2^4}$ is taken as example and the irreducible polynomial $f(z) = z^4 + z + 1$ is used to generate the field. An element $a_3z^3 + a_2z^3 + a_1z + a_0 \in \mathbb{F}_{2^4}, a_i \in \{0,1\}$, can be represented as bit strings. Let $a = a_3z^3 + a_0$ and $b = a_3z^3 + a_1z + a_0 \in \mathbb{F}_{2^4}, a_i \in \{0,1\}$.

$$E : y^2 + xy = x^3 + (a_3z^3 + a_0)x^2 + (a_3z^3 + a_1z + a_0) \tag{2}$$

$$E : y^2 + xy + x^3 + (z^3 + 1)x^2 + (z^3 + z + 1) = 0 \tag{3}$$

We look values for x and $y \in \mathbb{F}_{2^4}$ such that satisfies the equation 1.
If we try for example $x = z^2 + z \in \mathbb{F}_{2^m}$, and we replace it in the equation 3

$$E : y^2 + (z^2 + z)y + (z^2 + z)^3 + (z^3 + 1)(z^2 + z)^2 + (z^3 + z + 1) = 0$$

$$E : 1 + y^2 + z + yz + z^2 + yz^2 + 4z^3 + 4z^4 + 4z^5 + 3z^6 + z^7 = 0$$

We apply reduce $f(z) = z^4 + z + 1$ and module 2.

$$E : 1 + y^2 + z + yz + z^2 + yz^2 + 4z^3 + 4z^4 + 4z^5 + 3z^6 + z^7 \ mod\ \{p(z), 2\} = 0$$

We obtain the following equation:

$$E' : y^2 + yz + yz^2 = 0$$

You can use the following script in Wolfram Cloud [8] to determine and check the result:

```
Input:
Eq = y^2 + (z^2 + z)y + (z^2 + z)^3 + (z^3 + 1)(z^2 + z)^2 + (z^3 +z + 1)
(* Eq = 1 + y^2 + z + yz + z^2 + yz^2 + 4z^3 + 4z^4 + 4z^5 + 3z^6 + z^7 *)
IrreduciblePolynomialCCE = z^4 + z + 1
X = PolynomialMod[(Eq), {IrreduciblePolynomialCCE}];
X = PolynomialMod[(X), {2}];
Print [X]

Output:
y^2 + yz + yz^2
```

We now test for example the value of $y = z^2 + z \in \mathbb{F}_{2^m}$

$$E' : (z^2 + z)^2 + (z^2 + z)z + (z^2 + z)z^2 = 0$$

$$E' = 2z^2 + 4z^3 + 2z^4 = 0$$

We apply reduce module $f(z) = z^4 + z + 1$ and module 2.

$$2z^2 + 4z^3 + 2z^4 \ mod\ \{p(z), 2\} = 0$$

And finally we get: $E : 0 = 0$ that satisfies the equation 3.

Therefore the point $P = (z^2 + z,\ z^2 + z)$ belongs to the curve

$$E : y^2 + xy + x^3 + (z^3 + 1)x^2 + (z^3 + z + 1).$$

Since $P = (z^2 + z,\ z^2 + z)$ has a binary representation $P = (0110,\ 0110)$ and in turn in decimal representation, we would have $P = (6, 6)$.

Important note: In the rest of the document, we use the decimal notation as $P = (x, y)_{10}$ or specifying it in the description to facilitate this reading for the points of the curve E without forgetting that these points are actually polynomials of the form: $a_i z^i + a_{i-1} z^{i-1} + ...a_1 z + a_0 \in \mathbb{F}_{2^m}$, $a_i \in 0, 1$, with $0 \geq i \leq m - 1$ and $m \geq 1$.

If we calculate all the points that belong to curve 3 in its decimal notation we obtain:

(0, 14)	(4, 13)	(8, 10)	(12, 13)
(1, 12)	(5, 0)	(10, 0)	(14, 1)
(1, 13)	(5, 5)	(10, 10)	(14, 15)
(3, 4)	(6, 0)	(11, 1)	ϑ
(3, 7)	(6, 6)	(11, 10)	
(4, 9)	(8, 2)	(12, 1)	

Definition The order of an elliptic curve E defined on the finite field $\mathbb{F}_q$ is the number of points that exist on the curve and is denoted by $\#E(\mathbb{F}_q)$ [9].

In this case we have $\#E(\mathbb{F}_{2^m}) = 22$ of the elliptic curve E 3.

Group of laws for $E/\mathbb{F}_{2^m}$ of the form $y^2 + xy = x^3 + ax^2 + b$

As in the real fields, in the finite fields $\mathbb{F}_{2^m}$ we can define the operations that are used in the following way [10]:

- Identity: For all $P \in E(\mathbb{F}_{2^m})$ we have that $P + \vartheta = \vartheta + P = P$.
- Negative: If $P = (x, y) \in E(\mathbb{F}_{2^m})$, then $(x,\ y) + (x,\ x + y) = \vartheta$. The point $(x,\ x + y)$ is denoted by $-P$ and it is called the negative of P.
- Point addition: Let $P = (x_1, y_1)$ and $Q = (x_2, y_2) \in E(\mathbb{F}_{2^m})$ y $P \neq {}^{\pm}Q$. Then $P + Q = (x_3,\ y_3)$, where:

$$x_3 = \lambda^2 + \lambda + x_1 + x_2 + a$$
$$y_3 = \lambda(x_1 + x_3) + x_3 + y_1$$
$$\lambda = \frac{y_1 + y_2}{x_1 + x_2}$$

- Point doubling: Let $P = (x_1,\ y_1) \in E(\mathbb{F}_{2^m})$ where $P \neq -P$. Then $2P = (x_3,\ y_3)$, where:

$$x_3 = \lambda^2 + \lambda + a = x_1^2 + \frac{b}{x_1^2}$$
$$y_3 = x_1^2 + \lambda x_3 + x_3$$
$$\lambda = x_1 + \frac{y_1}{x_1}$$

Using the arithmetic of elliptic curves on finite fields $\mathbb{F}_{2^m}$ (using decimal notation) we have that: $(10,\ 10) + (3,\ 7) = (4,\ 9)$, Figure 7.

We can also observe that if for example we take the point $P = (14,\ 15)$ and applying the equations for double of a point, then we have that $2P = 2(14,\ 15) = (10,\ 0)$, and we can also observe that the point $P = (8,\ 10)$ has as negative point $-P = (8,\ 2)$ which is the reflection of P.

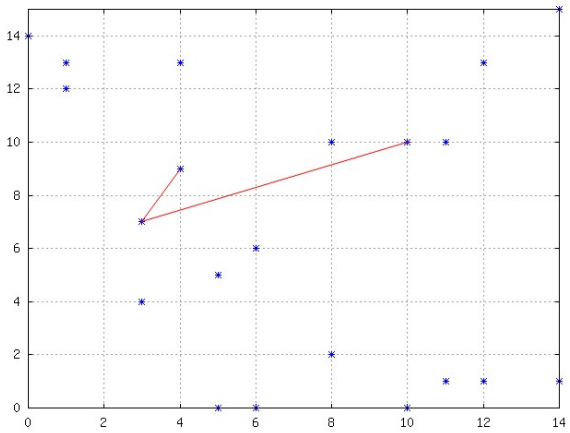

Figure 7: Graph of binary finite field $\mathbb{F}_{2^4}$ with the elliptic curve 3

Scalar multiplication

Definition If k is a positive integer and $P \in \mathbb{F}_{2^m}$, then $kP = \sum_{i=1}^{k} P$, denote the point obtained by adding k time the point P to itself. This process of calculating kP from P and k is called scalar multiplication [11].

Once the addition and doubling of a point for $E(\mathbb{F}_{2^m})$ have been defined, scalar multiplication can be performed on elliptic curves. We can use Algorithm 1 [4] to perform the multiplication in binary form. This algorithm is very efficient in binary representation. It is based on the idea of multiplying k times the point P, the binary representation of the scalar k is taken and a sweep is made from left to right, in such a way that in each step the doubling of point is performed and if the position of the current bit k_i is 1 then the addition operation of the current point is performed. Example:

$$15P = 2P + P \mapsto 6P + P \mapsto 14P + P \mapsto 15P$$

Algorithm 1 Addition and Doubling

Require: $k = (k_{j-1} \ldots k_1, k_0)$, $P \in \mathbb{F}_{2^m}$
Ensure: $R = kP$
1: $R \leftarrow P$
2: **for** $i = j - 1$ **until** 0 **do**
3: $R \leftarrow 2R$
4: **if** $k_j = 1$ **then**
5: $R = R + P$
6: **end if**
7: **end for**
8: **return** R

Summary:

In summary, to perform the **point addition** on the curve, we can use the equation established in the group of laws for elliptic curves of the form $E : y^2 + xy = x^3 + ax^3 + b$ defined on the finite field $\mathbb{F}_{2^m}$:
$(P_x, P_y) + (Q_x, Q_y) = (R_x, R_y)$, where P, Q y $R \in E(\mathbb{F}_{2^m})$:

$$R_x = \lambda^2 + \lambda + P_x + Q_x + a$$

$$R_y = \lambda(P_x + R_x) + R_x + P_y$$

$$\lambda = \frac{P_y + Q_y}{P_x + Q_x}$$

And to make the **point doubling** $(R_x,\ R_y) = 2(P_x,\ P_y)$ on the elliptic curve E over the finite field $\mathbb{F}_{2^m}$ we will use the following equation:

$$R_x = \lambda^2 + \lambda + a$$

$$R_y = {P_x}^2 + \lambda R_x + R_x$$

$$\lambda = P_x + \frac{P_y}{P_x}$$

And it is requested as a restriction that the coordinate $P_x \neq 0$.

Using the Algorithm 1 and with the help of *Wolfram Cloud* [8] we can develop a small script that performs scalar multiplication, where m represents the finite field binary $\mathbb{F}_{2^m}$, k is the integer to multiply in binary representation, a is the value within the Weierstrass equation, (P_x, P_y) is a point on the curve, and we must have an irreducible polynomial over $\mathbb{F}_{2^m}$. In this case the value of b not matter due to the equation selected above for addition and doubling of points. In this way, for example if we replace the following values into the Weierstrass equation, we have the following example:

$$E : y^2 + xy = x^3 + (z^3)x^2 + (b)$$

Where $a = z^3 \in \mathbb{F}_{2^4}$. The finite fiels binary $\mathbb{F}_{2^4}$ represented by the reduction polynomial $f(z) = z^4 + z + 1$, a point on the curve $P = (z^2 + z + 1,\ z^3 + z + 1) = (0111, 1011)_2 = (7, 11)_{10}$, with $k = 9$ the script would be the following:

```
Input:
(* GF(2^4), m = 4; *)
k = "1001"; (* Binary representation 1001 = 9 in decimal *)
lim = StringLength [k] + 1;
a = z^3;
Px = z^2 + z + 1;
Py = z^3 + z + 1;
IrreduciblePolynomialCCE = z^4 + z + 1;

Rx = Px;
Ry = Py;

For [i=2, i<lim , i++,
        c = StringTake [k ,{i}];
        (* Point doubling*)
        {d, {inv, u}} = PolynomialMod[PolynomialExtendedGCD[Rx, IrreduciblePolynomialCCE],2];
        Lamda = PolynomialMod[(Rx + Ry * inv ), {IrreduciblePolynomialCCE, 2}];
        xTmp = PolynomialMod[(Lamda^2 + Lamda + a ), {IrreduciblePolynomialCCE, 2}];
        yTmp = PolynomialMod [ (Rx^2 + Lamda * xTmp + xTmp) , {IrreduciblePolynomialCCE, 2}];
        Rx = xTmp;
        Ry = yTmp;
        If [c=="1",{
                        (* Point addition *)
                        {d, {inv2, u}} = PolynomialMod[ PolynomialExtendedGCD[Px + Rx, IrreduciblePolynomialCCE],2];
                        Lamda2 = PolynomialMod[(Py + Ry) * inv2 , {IrreduciblePolynomialCCE, 2}];
                        x3 = PolynomialMod [(Lamda2^2 + Lamda2 + Px + Rx + a) , {IrreduciblePolynomialCCE, 2}];
                        y3 = PolynomialMod [Lamda2 * (Px + x3) + x3 + Py, {IrreduciblePolynomialCCE, 2}];
                        Rx = x3;
                        Ry = y3;
                },{0}
        ]
]
Print [Rx]
Print [Ry]

Output:
Rx = z
Ry = z^3 + z^2 + z + 1
```

Where can see that $R_x = z = 0010_2 = 2_{10}$ and $R_y = z^3 + z^2 + z + 1 = 1111_2 = 15_{10}$ which means that $9(7, 11)_{10} = (2, 15)_{10}$ and this new point also lives on the curve.

Theorem Hasse: Let E be an elliptic curve defined over $\mathbb{F}_q$, Then $q+1-2\sqrt{q} \leq \#E(\mathbb{F}_q) \leq q+1+2\sqrt{q}$. The interval $[q+1-2\sqrt{q}, q+1+2\sqrt{q}\,]$ is called the Hasse interval. An alternate formulation of Hasse's theorem is the following: If E is defined over $\mathbb{F}_q$ then $\#E(\mathbb{F}_q) = q+1-t$ where $|\,t\,| \leq 2\sqrt{q}$; t is called the *trace* of E over $\mathbb{F}_q$. Since $2\sqrt{q}$ is small relative to q, we have $\#E(\mathbb{F}_q) \approx q$ [9].

Definition The order of a point is the least number of times that a point P must be added with itself to obtain the zero of the curve E. Basically it is to find the smallest number k such that $kP = \vartheta$, where ϑ is the identity of the curve $E(\mathbb{F}_{2^m})$. The order of any point always divides the order $\#E(\mathbb{F}_{2^m})$ of the curve [12].

Example group structure: Consider $\mathbb{F}_{2^7}$ as represented by the reduction polynomial $f(z) = z^7 + z + 1$. The elliptic curve $E : y^2 + xy = x^3 + 0 * x^2 + 1$ defined over $\mathbb{F}_{2^7}$ has $\#E(\mathbb{F}_{2^7}) = 116$. The points in $E(\mathbb{F}_{2^7})$ in its decimal representation are the following:

(0, 1)	(33, 27)	(48, 42)	(76, 126)	(89, 63)	(108, 123)
(1, 0)	(33, 58)	(50, 111)	(76, 50)	(90, 27)	(108, 23)
(1, 1)	(34, 67)	(50, 93)	(79, 46)	(90, 65)	(109, 12)
(12, 17)	(34, 97)	(53, 121)	(79, 97)	(91, 101)	(109, 97)
(12, 29)	(35, 105)	(53, 76)	(80, 7)	(91, 62)	(111, 57)
(13, 37)	(35, 74)	(56, 127)	(80, 87)	(94, 117)	(111, 86)
(13, 40)	(38, 35)	(56, 71)	(81, 88)	(94, 43)	(114, 57)
(15, 48)	(38, 5)	(58, 51)	(81, 9)	(98, 10)	(114, 75)
(15, 63)	(39, 108)	(58, 9)	(82, 100)	(98, 104)	(115, 51)
(25, 67)	(39, 75)	(59, 120)	(82, 54)	(100, 33)	(115, 64)
(25, 90)	(40, 19)	(59, 67)	(85, 30)	(100, 69)	(117, 105)
(26, 45)	(40, 59)	(61, 124)	(85, 75)	(102, 115)	(117, 28)
(26, 55)	(41, 121)	(61, 65)	(86, 105)	(102, 21)	(119, 37)
(28, 101)	(41, 80)	(66, 11)	(86, 63)	(103, 38)	(119, 82)
(28, 121)	(45, 114)	(66, 73)	(87, 8)	(103, 65)	(122, 37)
(29, 36)	(45, 95)	(71, 101)	(87, 95)	(104, 113)	(122, 95)
(29, 57)	(46, 112)	(71, 34)	(88, 3)	(104, 25)	(123, 27)
(30, 66)	(46, 94)	(73, 122)	(88, 91)	(107, 9)	(123, 96)
(30, 92)	(48, 26)	(73, 51)	(89, 102)	(107, 98)	(124, 56)
(124, 68)	ϑ				

Since 116 does not have any repeated factors, $E(\mathbb{F}_{2^7})$ is cyclic. The point $P = (z^6+z^3+z^2, z^6+z^5+z^4+z^3+z^2+z) = (1001100, 1111110) = (76, 126)$ has order $n = 29$ that divide $\#E(\mathbb{F}_{2^7}) = 116$ and its multiples in its decimal representation are shown below.

1(76, 126)	8(80, 87)	15(108, 123)	22(48, 26)	29 ϑ
2(102, 21)	9(30, 66)	16(124, 68)	23(98, 10)	30(76, 126)
3(12, 17)	10(88, 91)	17(38, 5)	24(82, 54)	31(102, 21)
4(46, 94)	11(40, 19)	18(40, 59)	25(46, 112)	32(12, 17)
5(82, 100)	12(38, 35)	19(88, 3)	26(12, 29)	...
6(98, 104)	13(124, 56)	20(30, 92)	27(102, 115)	
7(48, 42)	14(108, 23)	21(80, 7)	28(76, 50)	

Koblitz elliptic curves

The Koblitz elliptic curves, also known as anomalous binary curves are defined on $\mathbb{F}_2$ by the equations [18]:

$$E_0 : y^2 + xy = x^3 + 1$$

$$E_1 : y^2 + xy = x^3 + x^2 + 1$$

This type of curves known as E_a generically where $a = 0$ or $a = 1$. The group of points of the elliptic curve are

generated over an extension k of $\mathbb{F}_{2^m}$ and this group is denoted $E_a(\mathbb{F}_{2^m})$. Let l be a divisor of m. Group $E_a(\mathbb{F}_{2^l})$ is a subgroup of $E_a(\mathbb{F}_{2^m})$, therefore order $\#E_a(\mathbb{F}_{2^l})$ divides order $\#E_a(\mathbb{F}_{2^m})$. Also $\#E_0(2) = 4$ and $\#E_1(2) = 2$, therefore the order of any point of curves E_a generated on $\mathbb{F}_{2^m}$ is a multiple of 4 in curves E_0 and multiple of 2 in curves E_1.

A number $\#E(\mathbb{F}_{2^m}) = hn$ is called *almost-prime* if it can be factored with a prime number n and a relatibely small number h with $h \in 2,3,4$. Consider group $E_a(\mathbb{F}_{2^m})$ where m is a prime number. If order $\#E_a(\mathbb{F}_{2^m})$ is almost-prime, so this can factored as $\#E_a(\mathbb{F}_{2^m}) = hn$ where n is prime and h is called the *cofactor* and is:

- $h = 4$ if $a = 0$.
- $h = 2$ if $a = 1$.

Example:
Let the finite field $\mathbb{F}_{2^7}$ be formed by the irreducible polynomial $f(z) = z^7 + z + 1$ and the curve formed by the equation $E_0 : y^2 + xy = x^3 + 1$ where $a = 0$, and we know that $h = 4$. The point $P = (76,126)_{10}$ on the curve with a prime number $n = 29$, then we have: the number $\#E_0(\mathbb{F}_{2^7}) = hn = (4)(29) = 116$.

So then, the binary finite field $\mathbb{F}_{2^{41}}$ formed by the irreducible polynomial $f(z) = z^{41} + z^3 + 1$ and the curve formed by the equation $E_0 : y^2 + xy = x^3 + 1$ where $a = 0$, and we know that $h = 4$. The point $P = (1001228071526, 1146921837697)_{10}$ on the curve with a prime number $n = 549756390943$, then we can determine that $\#E_0(\mathbb{F}_{2^{41}}) = hn = (4)(549756390943) = 2199025563772$.

Elliptic Curve Discrete Logarithm Problem (ECDLP)

Let E be an elliptic curve over a finite field and let G be a point on E of order n. The Elliptic Curve Discrete Logarithm Problem (ECDLP) is: Given E and G and a scalar multiple Q of G, determine an integer $k < n - 1$ such as $Q = kG$ [13].

Elliptic-Curve Diffie-Hellman (ECDH) protocol

Elliptic-Curve Diffie-Hellman (ECDH) is a key agreement protocol that allows two entities, to establish a share secret over an insecure channel like the Internet.

The protocol elliptic Diffie-Hellman over the finite field $\mathbb{F}_{2^m}$ for curve $E : y^2 + xy = x^3 + ax^2 + b$, where a and b are elements of $\mathbb{F}_{2^m}$, the steps for exchanging keys are as follows [14]:

A binary finite field $\mathbb{F}_{2^m}$ is agreed for entities A and B, through the value m with the irreducible polynomial $f(z)$, the values of $a, b \in \mathbb{F}_{2^m}$ over the curve $E : y^2 + xy = x^3 + ax^2 + b$ and a point G that lies on E, the point G has order n and the cofactor h.

Entity A

- Generate a random integer value k_a (Private key), where $k_a \in [1, n-1]$.
- Calculate $Q = [k_a]G$.
- Send point Q (Public key).
- Receive R.
- Calculate $S = [k_a]R = [k_a][k_b]G$.
- Encrypt the message using S.
- Send the encrypted message.

Entity B

- Generate a random integer value k_b (Private key), where $k_b \in [1, n-1]$.
- Calculate $R = [k_b]G$.
- Send point R (Public key).
- Receive Q.
- Calculate $S = [k_b]Q = [k_b][k_a]G$.
- Receive the encrypted message.
- Decrypt the message using S.

Entity C

- Receive the values of: $m, f(z), a, b, G, n, h$.
- Receive Q (Public key).
- Receive R (Public key).
- Receive the encrypted message.

So that C can decrypt the message, need to calculate k_a and k_b facing the ECDLP, Figure 8.

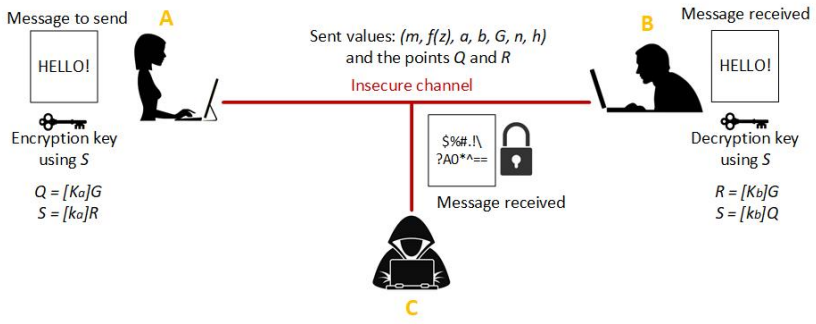

Figure 8: Elliptic-Curve Diffie-Hellman (ECDH) protocol, for key exchange.

Example of ECDH protocol using decimal notation for the points that live on the given E curve:

Suppose we have the curve $E : y^2 + xy = x^3 + ax^2 + b$ over the finite field $\mathbb{F}_{2^m}$ and where a and b are elements of $\mathbb{F}_{2^m}$, we have the following values:

A binary finite field is agreed for entities A and B, generating the values of $m = 7$, $a = 0$, $b = 1$ and a point $G = (46, 112)$ that lies on E and the order of point G is $n = 29$.

Entity A

- Generate a random integer value $k_a = 19$ (Private key).
- Calculate $Q = [k_a]G = [19](46, 112) = (40, 19)$.
- Send point $Q = (40, 19)$ (Public key).
- Receive $R = (88, 91)$.
- Calculate $S = [k_a]R = [19](88, 91)$.
- Encrypt the message using $S = (124, 68)$.
- Send the encrypted message.

Entity B

- Generate a random integer value $k_b = 12$ (Private key).
- Calculate $R = [k_b]G = [12](46, 112) = (88, 91)$.
- Send point $R = (88, 91)$ (Public key).
- Receive $Q = (40, 19)$.
- Calculate $S = [k_b]Q = [12](40, 19) = (124, 68)$.
- Receive the encrypted message.
- Decrypt the message using $S = (124, 68)$.

Entity C

- Receive the values of: $m = 7, f(z), a = 0, b = 1, G = (46, 112), n = 29, h = 4$.
- Receive $Q = (40, 19)$ (Public key).
- Receive $R = (88, 91)$ (Public key).
- Receive the encrypted message.

So that C can decrypt the message, need to calculate k_a and k_b facing the ECDLP.

Encryption and decryption of messages using the ElGamal Elliptic Curve Cryptosystem

The public key cryptography or asymmetric cryptography uses two keys, public and private key. Unlike private key cryptography that use the same two keys during encryption and decryption process, the public key cryptography, does not require any shared secret between the communicating parties.
The next section summarizes how you could create a cryptosystem for encryption and decryption of messages using elliptic curves cryptosystem and ElGamal algorithm [15]. If we want to send message, first we divide the message M into M_n blocks, now it is necessary to apply a function such that a block of the plaintext message encoded be $M_n \mapsto P_m$ and P_m is a point that belongs to the curve $E : y^2 + xy = x^3 + ax^2 + b$, where a and b are elements of $\mathbb{F}_{2^m}$.

The steps for encryption and decryption messages are as follows:
A binary finite field is agreed for entities A and B, generating the values of m with the irreducible polynomial $f(z)$, the values of a, $b \in \mathbb{F}_{2^m}$ and a point G that is the generator point and lies on E and the point G has order n and the cofactor h.

Encryption

- Entity A gets B's public key (R).
- Entity A apply a function in plaintext message block to encode M_n and get point P_m that lies on the curve E.
- Entity A selects a random number $k_a \in [1, n-1]$ calculate $Q = [k_a]G$.
- Entity A calculate $E_M = ([k_a]R) + P_m$ that will be the encrypted message.
- Entity A sends $\omega = (Q, E_M)$ that is the ciphertext pair of points for the plaintext point P_m and is send to entity B.

Decryption

- Entity B receive the value $\omega = (Q, E_M)$.
- Entity B calculate $S = [k_b]Q$ where $k_b \in [1, n-1]$ is his private key.
- Entity B calculate $-S$ using the negative of group of laws for $E/\mathbb{F}_{2^m}$.
- Entity B calculate $P_m = E_M - S$ where P_m is the representation of message encoded.
- Entity B decode P_m such that $P_m \mapsto M_n$ where M_n is the plaintext message of the corresponding block.

Example of ElGamal Elliptic Curve Algorithm using decimal notation for the points that live on the given E curve:

Suppose we have the curve $E : y^2 + xy = x^3 + ax^2 + b$ over the finite field $\mathbb{F}_{2^m}$ and where a and b are elements of $\mathbb{F}_{2^m}$, we have the following values:

A binary finite field is agreed for entities A and B, generating the values of $m = 7$, $a = 0$, $b = 1$ and a point $G = (40, 19)$ that is the generator point and lies on E the point G has order $n = 29$.

Encryption

- Entity A gets B's public key $R = (88, 91)$.
- Entity A apply a function in plaintext message block to encode M_n and get point and assuming it is $P_m = (107, 98)$ that lies on the curve E.
- Entity A selects a random number $k_a = 19$ calculate $Q = [k_a]G = [19](46, 112) = (40, 19)$.
- Entity A calculate $E_M = [k_a]R + P_m = [19](88, 91) + (107, 98) = (124, 68) + (107, 98)$, $EM = (117, 105)$ that will be the encrypted message.
- Entity A sends $\omega = [Q, E_M] = [(40, 19), (117, 105)]$ that is the ciphertext pair of points for the plaintext point P_m and is send to entity B.

Decryption

- Entity B receive the value $\omega = [Q, E_M] = [(40, 19), (117, 105)]$.
- Entity B calculate $S = [k_b]Q = [12](40, 19) = (124, 68)$.
- Entity B calculate $-S$ with $S = (124, 68)$, $-S = (124, 56)$.
- Entity B calculate $P_m = E_M - S = (117, 105) + (124, 56) = (107, 98)$ where P_m is the representation of message encoded.
- Entity B decode $P_m = (107, 98)$ such that $P_m \mapsto M_n$ where M_n is the plaintext message of the corresponding block.

Elliptic Curve Digital Signature Algorithm (ECDSA) - Signature Generation and Verification

The digital signature allows the recipient of a message to verify the authenticity of the origin of the information as well as to verify that said information has not been altered since its generation. In this way, the digital signature offers support for the authentication and integrity of the information as well as for non-repudiation at origin, since the originator of a digitally signed message cannot say that it is not. We can summarize the Digital Signature Algorithm (DSA) as shown in the figure 9.

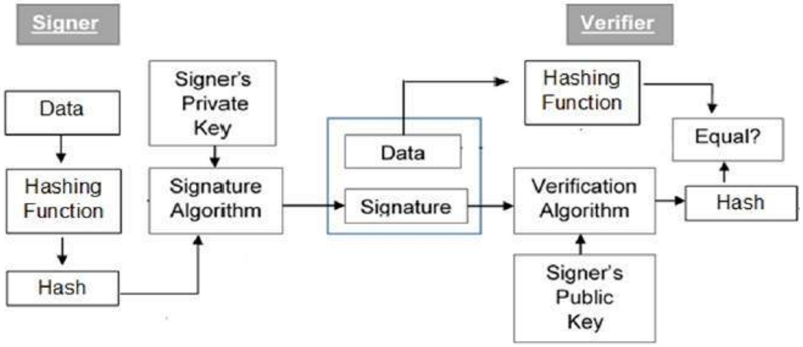

Figure 9: Block diagram of DSA. Image taken from: https://www.includehelp.com/cryptography/Images/dsa-1.jpg

The next section summarizes the procedures for generating and verifying signatures using Elliptic Curve Digital Signature Algorithm (ECDSA) [16].
To sign a message M, a binary finite field is agreed for entities A and B, generating the values of m with the irreducible polynomial $f(z)$, the values of a, $b \in \mathbb{F}_{2^m}$ and a point G that is the generator point and lies on E, the point G has order n, the cofactor h and associated key pair (d,Q) does the following:

ECDSA Signature generation

- Entity A select a random or pseudorandom integer $k, 1 \leq k \leq n-1$.
- Entity A compute $[k]G = (x_1, y_1)$ and convert x_1 to an integer $\bar{x}_1$
- Entity A compute $r = \bar{x}_1 \ mod \ n$. If $r = 0$ then go back to the initial step.
- Entity A compute $k^{-1} \ mod \ n$.
- Entity A compute SHA-1(M) and convert this bit string to an integer e.
- Entity A compute $s = k^{-1}(e + dr) \ mod \ n$. If $s = 0$ then go back to the initial step.
- A's signature for the message M is (r,s).

To verify A's signature (r,s) on M, B obtains an authentic copy of A's parameters $(m, f(z), a, b, G, n, h)$ and associated public key Q.

ECDSA Signature Verification

- Entity B verify that r and s are integers in the interval $[1, n-1]$.
- Entity B compute SHA-1(M) and convert this bit string to an integer e.
- Entity B compute $w = s^{-1} \ mod \ n$,
- Entity B compute $u_1 = ew \ mod \ n$ and $u_2 = rw \ mod \ n$.
- Entity compute $X = [u_1]G + [u_2]Q$.
- If $X = \vartheta$, then reject the signature. Otherwise, convert the x-coordinate x_1 of X to an integer $\bar{x}_1$, and compute $v = \bar{x}_1 \ mod \ n$.
- Entity B accept the signature if and only if $v = r$.

Example of ECDSA using decimal notation for the points that live on the given E curve:

Suppose we have the curve $E : y^2 + xy = x^3 + ax^2 + b$ over the finite field $\mathbb{F}_{2^m}$, to sign a message M,a binary finite field is agreed for entities A and B, generating the values of $m = 7$ with the irreducible polynomial $f(z) = z^7 + z + 1$, the values of $a = 0$, $b = 1$ and a point $G = (46, 112)$ that is the generator point and lies on E, the point G has order $n = 29$, the cofactor $h = 4$, A's private key $d = 19$, the public key $Q = [d]G = (40, 19)$ that is a point and lies on E.

ECDSA Signature generation

- Entity A select a random or pseudorandom integer $k = 22, 1 \leq k \leq 29-1$.
- Entity A compute $[k]G = (76,50)$ and convert 76 to an integer $\overline{76}$
- Entity A compute $r = \overline{76}\ mod\ 29 = 18$.
- Entity A compute $22^{-1}\ mod\ 29 = 4$.
- Entity A compute SHA-1(M) and convert this bit string to an integer for example $e = 27$.
- Entity A compute $s = k^{-1}(e+dr)\ mod\ n = 4(27+19*18)\ mod\ 29 = 26$. If $s = 0$ then go back to the initial step.
- A's signature for the message M is $(r,s) = (18,26)$.

To verify A's signature $(r,s) = (18,26)$ on M, B obtains an authentic copy of A's parameters $(m = 7, f(z), a = 0, b = 1, G = (46,112), n = 29, h = 4)$ and associated public key $Q = (40,19)$.

ECDSA Signature Verification

- Entity B verify that $r = 18$ and $s = 26$ are integers in the interval $[1, 29-1]$.
- Entity B compute SHA-1(M) and convert this bit string to an integer $e = 27$.
- Entity B compute $w = 26^{-1}\ mod\ 29 = 19$,
- Entity B compute $u_1 = (27)(19)\ mod\ 29 = 20$ and $u_2 = (18)(19)\ mod\ 29 = 23$.
- Entity compute $X = [u_1]G + [u_2]Q = [20](46,112) + [23](40,19) = (48,42) + (80,7) = (76,50)$.
- If $X = \vartheta$, then reject the signature. Otherwise, convert the x-coordinate 76 of X to an integer $\overline{76}$, and compute $v = \overline{76}\ mod\ 29 = 18$.
- Entity B accept the signature if and only if $v = r$.

Elliptic curve parameters

In real environments, it is necessary to define adequate parameters on the elliptic curves reducing the chance of attack. The domain parameters of the elliptic curves are the basic values necessary to define the finite field to use, such as the a and b values that define the curve, etc. Although these can be generated by any entity such as the Certificate Authority (CA). These parameters must be shared by the parties that want to communicate, so that in general it is a matter of always using the same parameters recommended by the standards-producing organizations.

Elliptic curve parameters on $\mathbb{F}_{2^m}$

The domain parameters of the elliptic curve over $\mathbb{F}_{2^m}$ are a septuple [17]:

$$T = (m, f(z), a, b, G, n, h)$$

Consisting of an integer m that defines the finite field $\mathbb{F}_{2^m}$, an irreducible polynomial $f(z)$ of degree m, give explicitly two elements a, $b \in \mathbb{F}_{2^m}$ specifying the shape of the elliptic curve $E(\mathbb{F}_{2^m})$ defined by the equation:

$$y^2 + xy = x^3 + ax^2 + b \tag{4}$$

a base point $G = (x_G, y_G) \in E(\mathbb{F}_{2^m})$, a prime number n which is the order of G, and an integer h which is the cofactor $h = \#E(\mathbb{F}_{2^m})/n$.

The process to generate a septuple $T = (m, f(z), a, b, G, n, h)$ is as follows:

1. Select the appropriate level of security (in bits) for the elliptic curve parameters, this must be an integer $t \in \{56, 64, 80, 96, 112, 128, 192, 256\}$ so that calculating logarithms over the associated elliptic curve takes approximately 2^t operations.

2. Let t' be the smallest number that is greater that t in the set $\{64, 80, 96, 112,$ $128, 192, 256, 512\}$. Select $m \in \{113, 131, 163, 193, 233, 239, 283, 409, 571\}$ such that $2t < m < 2t'$ to determine the finite field $\mathbb{F}_{2^m}$.

3. Select an irreducible binary polynomial $f(z)$ of degree m from Table 1 to determine the representation of $\mathbb{F}_{2^m}$. This restriction is designed to facilitate interoperability between implementations.

4. Select elements $a, b \in \mathbb{F}_{2^m}$ to determine the elliptic curve $E(\mathbb{F}_{2^m})$ defined by the $E : y^2 + xy = x^3 + ax^2 + b \in \mathbb{F}_{2^m}$, a base point $G = (x_G, y_G) \in E(\mathbb{F}_{2^m})$, a prime number n which is the order of G, and an integer h, which is the cofactor $h = \#E(\mathbb{F}_{2^m})/n$, subject to the following restrictions:

   (a) $b \neq 0$ in $\mathbb{F}_{2^m}$.
   (b) $\#E(\mathbb{F}_{2^m}) \neq 2^m$.
   (c) $2^{mB} \neq 1 \pmod{n}$ for all $1 \leq B < 20$.
   (d) $h \leq 4$

5. Return $(m, f(z), a, b, G, n, h)$.

1. The restrictions that the parameters must meet are:

   (a) It is required to meet the definition of elliptic curves with respect to the discriminant Δ over $\mathbb{F}_{2^m}$.
   (b) It is required to avoid anomalous curves in which is strictly required that $\#E(\mathbb{F}_{2^m}) \neq 2^m$.
   (c) It is required to avoid supersingular curves
   (d) It is required that $h \leq 4$ so that $\#E(\mathbb{F}_{2^m})$ is a number close to a large prime number.

The rule for choosing the irreducible polynomial $f(z)$ is to choose a trinomial of the form $f(z) = z^m + z^k + 1$, with $m > k \geq 1$, and k as small as possible. If an irreducible trinomial does not exist, a pentanomial of the form$f(z) = z^m + z^{k_3} + z^{k_2} + z^{k_1} + 1$ is chosen, $m > k_3 > k_2 > k_1 > 1$, with k_3 as small as possible, k_2 as small as possible given k_3, y k_1 as small as possible given k_3 and k_2.

Field	**Irreducible polynomials**
$\mathbb{F}_{2^{163}}$	$f(z) = z^{163} + z^7 + z^6 + z^3 + 1$
$\mathbb{F}_{2^{233}}$	$f(z) = z^{233} + z^{74} + 1$
$\mathbb{F}_{2^{283}}$	$f(z) = z^{283} + z^{12} + z^7 + z^5 + 1$
$\mathbb{F}_{2^{409}}$	$f(z) = z^{409} + z^{87} + 1$
$\mathbb{F}_{2^{571}}$	$f(z) = z^{571} + z^{10} + z^5 + z^2 + 1$

Table 1: NIST-Recommended polynomials in FIPS 186-3 [19]

References

[1] R. McEliece *Finite Fields for Computer Scientists and Engineers*. Springer.

[2] Stanford University: Chapter 7, Introduction to finite fields,
http://web.stanford.edu/~marykw/classes/CS250_W19/readings/Forney_Introduction_to_Finite_Fields.pdf

[3] H.Kamarulhaili and K. Liew: Elliptic Curve Cryptography and Point Counting Algorithms,
https://www.researchgate.net/profile/Hailiza-Kamarulhaili/publication/221927172_Elliptic_Curve_Cryptography_and_Point_Counting_Algorithms/links/546ddaa90cf2193b94c5d342/Elliptic-Curve-Cryptography-and-Point-Counting-Algorithms.pdf

[4] N. Falah and M. Rushdan: High Performance Methods of Elliptic Curve Scalar Multiplication,
https://www.researchgate.net/profile/Najlae-Falah-Hameed-Saffar/publication/284494383_High_Performance_Methods_of_Elliptic_Curve_Scalar_Multiplication/links/5d0b2b25299bf1f539d1880f/High-Performance-Methods-of-Elliptic-Curve-Scalar-Multiplication.pdf

[5] L. Mun-Kyu, K. Keon, K. Howon and K. Dong: Efficient Hardware Implementation of Elliptic Curve Cryptography over $GF(p^m)$,
https://doi.org/10.1007/11604938_16

[6] D. Hankerson, A. Menezes and S. Vanstone *Guide to Elliptic Curve Cryptography*. Springer-Verlag, pp.78-79, 2003.

[7] K. Hakuta: Metrics on the Sets of Nonsupersingular Elliptic Curves in Simplified Weierstrass Form over Finite Fields of Characteristic Two,
https://cutt.ly/4lTBgx7

[8] Wolfram Cloud,
https://lab.open.wolframcloud.com/objects/wpl/GetStarted.nb

[9] D. Hankerson, A. Menezes and S. Vanstone *Guide to Elliptic Curve Cryptography*. Springer-Verlag, pp. 82, 2003.

[10] D. Hankerson, A. Menezes and S. Vanstone *Guide to Elliptic Curve Cryptography*. Springer-Verlag, pp. 81, 2003.

[11] A Run Time Reconfigurable Co-Processor for Elliptic Curve Scalar Multiplication,
https://ccc.inaoep.mx/~cferegrino/Publicaciones/articulos/ReconfCoprocECC.pdf

[12] Taylor & F. Group: Elliptic Curves over Finite Fields,
http://koclab.cs.ucsb.edu/teaching/ecc/eccPapers/Washington-ch04.pdf

[13] M. Jacobson, A. Menezes and A. Stein: Solving Elliptic Curve Discrete Logarithm Problems Using Weil Descent,
https://www.math.uwaterloo.ca/~ajmeneze/publications/124.pdf

[14] R. Haakegaard and J. Lang: The Elliptic Curve Diffie-Hellman (ECDH),
http://koclab.cs.ucsb.edu/teaching/ecc/project/2015Projects/Haakegaard+Lang.pdf

[15] M. Saikia: Implementation of ElGamal Elliptic Curve Cryptography over prime field using C,
https://www.researchgate.net/publication/272162532_Implementation_of_ElGamal_Elliptic_Curve_Cryptography_over_prime_field_using_C

[16] D.Johnson, A. Menezes, and S. Vansto: The Elliptic Curve Digital Signature Algorithm (ECDSA),
https://www.cs.miami.edu/home/burt/learning/Csc609.142/ecdsa-cert.pdf

[17] Certicom Research, Elliptic Curve Cryptography, Standards for Efficient Cryptography, https://www.secg.org/SEC1-Ver-1.0.pdf

[18] D. Hankerson, A. Menezes and S. Vanstone *Guide to Elliptic Curve Cryptography*. Springer-Verlag, pp. 114, 2003.

[19] FIPS PUB 186-3: U.S. DIGITAL SIGNATURE STANDARD (DSS), https://csrc.nist.gov/csrc/media/publications/fips/186/3/archive/2009-06-25/documents/fips_186-3.pdf