Elliptic Curve Cryptography

Gerardo Pelosi

Dipartimento di Elettronica, Informazione e Bioingegneria (DEIB)
Politecnico di Milano

gerardo.pelosi - at - polimi.it

Elliptic Curve Cryptography (ECC)

- ECC was introduced by Victor Miller and Neal Koblitz in 1985
- their security is based on the DLP in a finite cyclic group composed by the points of an elliptic curve
- the EC-DLP has a fully exponential computational complexity. This means that assuming a certain security margin:
  - ECC requires significantly smaller key size w.r.t. RSA, or DL cryptoschemes defined over $\mathbb{F}_{p^m}$
- benefits of having smaller key sizes: faster computations, need less storage space
- ECCs are ideal for the constrained environments of embedded systems: e.g., pagers, PDAs, smart phones, smart cards

Elliptic Curve Cryptography (ECC)

Elliptic curve

Let $\mathbb{K}$ be a field. An elliptic curve over $\mathbb{K}$ is defined as the set of solutions $(x, y) \in \mathbb{K} \times \mathbb{K}$ of a (so-called) Weierstrass equation:

$$\mathbb{E}(\mathbb{K}) = \{(x, y) \in \mathbb{K}^2 : \ y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6, \ a_i \in \mathbb{K}\}$$

This curve has to be also non-singular. Geometrically, this means that the graph has no cusps, self-intersections, or isolated points. This is equivalent to say that every point of the curve should have a unique tangent line.

The definition is given for any field $\mathbb{K}$, however, for cryptographic applications only the finite fields are considered: $\mathbb{K} = \mathbb{F}_{2^m}$ or $\mathbb{K} = \mathbb{F}_{p^m}$, $p \geq 3$

Elliptic Curves

if $\mathbb{K} = \mathbb{F}_p, p > 3$, the Weierstrass equation can be reduced to a simpler form through the following change of variables

$$\mathbf{x} \leftarrow x - \frac{a_2}{3}$$
$$\mathbf{y} \leftarrow y - \frac{a_1x + a_3}{2}$$

subsequently, through denoting $a = \frac{1}{9}a_1^2 + a_4$, $b = \frac{2}{27}a_2^3 - \frac{1}{3}a_2a_4a_6$ we obtain

$$\mathbb{E}(\mathbb{F}_p) : \ \mathbf{y}^2 = \mathbf{x}^3 + a\mathbf{x} + b \quad a, b \in \mathbb{F}_p, \ p > 3$$

Denoting the curve as $y^2{=}f(x)$, and considering its derivative $f'(x){=}2y\frac{dy}{dx}$, we note that $\frac{dy}{dx}$ is indefinite when $f'(x_0){=}f(x_0){=}y_0{=}\,0$
Equivalently, the 3rd degree equation $f(x){=}x^3 + ax + b$ must not have multiple roots.
This forces the equation to have a non-zero discriminant in order to represent a non-singular curve:

$$\boldsymbol{\Delta = 4a^3 + 27b^2 \neq 0}$$

Elliptic Curves

if $\mathbb{K} = \mathbb{F}_{2^m}$, the Weierstrass equation can be reduced to a simpler form through the following change of variables, assuming $a_1 \neq 0$:

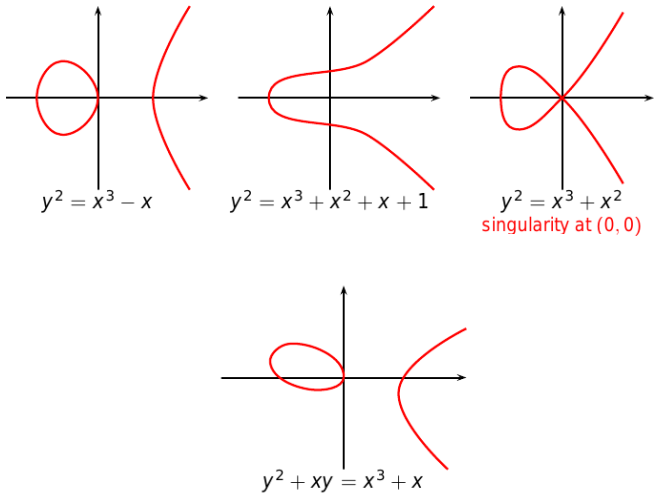
$$\mathbf{x} \leftarrow a_1^2 x - \frac{a_3}{a_1}$$
$$\mathbf{y} \leftarrow a_1^3 y - \frac{a_1^2 a_4 + a_3^2}{a_1^3}$$

subsequently, through denoting $a = \frac{1}{9}a_1^2 + a_4$, $b = \frac{2}{27}a_2^3 - \frac{1}{3}a_2 a_4 a_6$ we obtain

$$\mathbb{E}(\mathbb{F}_{2^m}) : \ \mathbf{y}^2 + \mathbf{xy} = \mathbf{x}^3 + a\mathbf{x}^2 + b \quad a, b \in \mathbb{F}_{2^m}, \ b \neq 0$$

The requirement of having no singularities in every point of the curve corresponds to the condition: $\mathbf{b} \neq \mathbf{0}$.

Elliptic Curves over $\mathbb{R}$



Elliptic Curve Cryptography (ECC)

In order to define a cryptosystem we need to find a proper algebraic structure for the set of points belonging to the curve

The simplest structure that gives sufficient properties to define a cryptosystem is the group structure $(G, +)$, where G is a set of curve points and the internal composition law (i.e., the $+$ operation) is properly defined.

Elliptic Curve Cryptography (ECC)

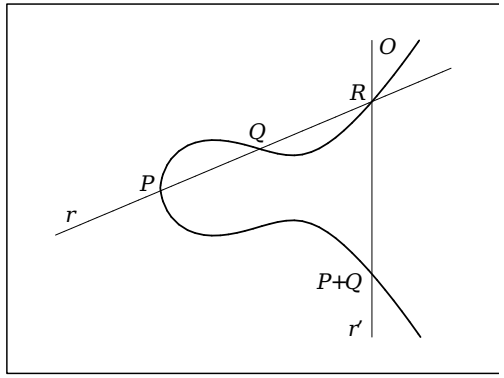
Denote with $\mathcal{O}$ the neutral element of the group, and assume (for now) $\mathcal{O}$ be a pointy at infinity along the vertical direction, parallel to the y-axis

Composition Law: Chord & Tangent Rule

Let $P, Q \in \mathbb{E}(\mathbb{K})$ be two points and $r = \overline{PQ}$ be the straight line passing through P and Q (or the tangent line if $P = Q$).
Denote with R the third point intercepted by the straight line $r = \overline{PQ}$ (Obs: the curve $y^2 = f(x)$, where $f(x)$ is a 3rd degree polynomial, will always intercept a straight line in at most 3 points).
Denote with $r' = \overline{R\mathcal{O}}$ the vertical line passing through R (and $\mathcal{O}$).

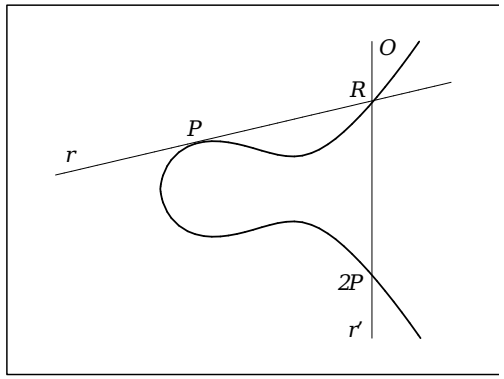
The sum of the two points P and Q, denoted by $S = P + Q$, is defined to be the third point intercepted by r' over $\mathbb{E}(\mathbb{K})$.

Chord & Tangent Rule: Point Addition $S=P+Q$



$$\mathbb{E}(\mathbb{R}):\ y^2 = x^3 - 3x + 6$$

Chord & Tangent Rule: Point Doubling $S=[2]P$



$$\mathbb{E}(\mathbb{R}):\ y^2 = x^3 - 3x + 6$$

Denoting with $\mathbb{K}$ a field. It is possible to show that the "Chord & Tangent Rule" gives to the points of any elliptic curve $\mathbb{E}(\mathbb{K})$ the structure of an **additive abelian group**, where the neutral element is represented by the point at infinity $\mathcal{O}$:

$$\forall P \in \mathbb{E}(\mathbb{K}),\ P + \mathcal{O} = P$$
$$\forall P \in \mathbb{E}(\mathbb{K}),\ \exists!\, Q = (-P) \in \mathbb{E}(\mathbb{K}) :\ P + Q = \mathcal{O}$$
$$\forall P, Q, R \in \mathbb{E}(\mathbb{K}),\ P + (Q + R) = (P + Q) + R$$
$$\forall P, Q \in \mathbb{E}(\mathbb{K}),\ P + Q = Q + P$$

When considering the finite fields ($\mathbb{K} = \mathbb{F}_{p^m}$), the curve points continue to have the same algebraic properties, although it is not possible to show a graphic representation of them (as we have done in the case of $\mathbb{K} = \mathbb{R}$)

Elliptic Curves over $\mathbb{F}_{p^m},\ p > 3$

Given the points $P_1 = (x_1, y_1),\ P_2 = (x_2, y_2)$, the explicit formulas to compute $P_1 + P_2 = (x_3, y_3)$ and $[2]P_1 = (x_4, y_4)$ are:

$P_1 + P_2$	$x_3 = (\frac{y_1 - y_2}{x_1 - x_2})^2 - x_1 - x_2$ $y_3 = (\frac{y_1 - y_2}{x_1 - x_2})(x_1 - x_3) - y_1$
$[2]P_1$	$x_4 = (\frac{3x_1^2 + a}{2y_1})^2 - 2x_1$ $y_4 = -y_1 + (\frac{3x_1^2 + a}{2y_1})(x_1 - x_4)$
$-P_1$	$(x_1, -y_1)$

Elliptic Curve Cryptography (ECC)

Elliptic Curves over $\mathbb{F}_{2^m}$

Given the points $P_1 = (x_1, y_1)$, $P_2 = (x_2, y_2)$, the explicit formulas to compute $P_1 + P_2 = (x_3, y_3)$ and $[2]P_1 = (x_4, y_4)$ are:

$P_1 + P_2$	$x_3 = (\frac{y_1+y_2}{x_1+x_2})^2 + (\frac{y_1+y_2}{x_1+x_2}) + x_1 + x_2 + a$ $y_3 = (\frac{y_1+y_2}{x_1+x_2})(x_1 + x_3) + x_3 + y_1$
$[2]P_1$	$x_4 = x_1^2 + \frac{b}{x_1^2}$ $y_4 = x_1^2 + (x_1 + \frac{y_1}{x_1})x_3 + x_3$
$-P_1$	$(x_1, x_1 + y_1)$

Elliptic Curve Cryptography (ECC)

The addition formulas are quite easy to remember if you consider the method to derive them:

Elliptic Curves over $\mathbb{F}_{p^m}$, $p > 3$

Given the points $P_1 = (x_1, y_1)$, $P_2 = (x_2, y_2)$, the coordinates of $P_1 + P_2 = (x_3, y_3)$ are derived from:

$$\begin{cases} y^2 = x^3 + ax + b \\ y = \lambda(x - x_1) + y_1, \lambda = \frac{y_1 - y_2}{x_1 - x_2} \text{ is the slope of the line } \overline{P_1 P_2} \end{cases}$$

substituting the 2nd eq. in the 1st one you can write $\mathbf{x^3} - \lambda^2\mathbf{x^2} + \ldots = \mathbf{0}$.
This equation has three solutions; two of them are known: x_1 and x_2.
The third one, x_3, can be quickly derived through remembering that "the sum of the roots of a monic polynomial with degree n, is equal to the opposite of the coefficient with degree $n-1$": $\mathbf{x_3} = \lambda^2 - \mathbf{x_1} - \mathbf{x_2}$.
Now using the 2nd eq. you find that $\mathbf{y_3} = \lambda(\mathbf{x_3} - \mathbf{x_1}) + \mathbf{y_1}$

Elliptic Curve Cryptography (ECC)

Elliptic Curves over $\mathbb{F}_{p^m}$, $p > 3$

The doubling formulas are derived in a similar fashion considering that the slope λ of the tangent line $y = \lambda(x - x_1) + y_1$ is computed as
$\lambda = y'(x_1) = \frac{3x_1^2 + a}{2y_1}$

Finally, the coordinate of the opposite of a given point are obtained through intersecting the curve with a vertical line ($x = x_1$)

Elliptic Curves over $\mathbb{F}_{2^m}$

In the case of binary finite fields, the method to derive the addition and doubling formulas is exactly the same method seen for curves over prime fields. (Pay attention to the fact that the arithmetic op.s are different)

From now on we will consider only curves over $\mathbb{F}_p$, $p > 3$, but the same conclusions can be obtained also for the binary fields.

Elliptic Curve Cryptography (ECC)

Example

Consider the curve $\mathbb{E}(\mathbb{F}_7)$: $y^2 = x^3 + x + 3$
It turns out there are six points on this curve (i.e., n=$|\mathbb{E}(\mathbb{F}_7)|$=6) given by

$$\mathcal{O},\ (4,1),\ (6,6),\ (5,0),\ (6,1),\ (4,6)$$

the group law can be specified through the following table:

+	$\mathcal{O}$	(4, 1)	(6, 6)	(5, 0)	(6, 1)	(4, 6)
$\mathcal{O}$	$\mathcal{O}$	(4, 1)	(6, 6)	(5, 0)	(6, 1)	(4, 6)
(4, 1)	(4, 1)	(6, 6)	(5, 0)	(6, 1)	(4, 6)	$\mathcal{O}$
(6, 6)	(6, 6)	(5, 0)	(6, 1)	(4, 6)	$\mathcal{O}$	(4, 1)
(5, 0)	(5, 0)	(6, 1)	(4, 6)	$\mathcal{O}$	(4, 1)	(6, 6)
(6, 1)	(6, 1)	(4, 6)	$\mathcal{O}$	(4, 1)	(6, 6)	(5, 0)
(4, 6)	(4, 6)	$\mathcal{O}$	(4, 1)	(6, 6)	(5, 0)	(6, 1)

Assume $P = (4, 1)$ then we have: $[2]P$=$(6, 6) \neq \mathcal{O}$, $[3]P$=$(5, 0) \neq \mathcal{O}$, thus $\mathbb{E}(\mathbb{F}_7)$ is a finite cyclic Abelian group of order six generated by the point P.
For all elliptic curves over finite fields, the group is always finite and it is also highly likely be cyclic (or "nearly" cyclic)

Elliptic Curves: Group Order

- Over a finite field $\mathbb{F}_q$ ($q = p^m$) the number of rational points on a curve, $n{=}|\mathbb{E}(\mathbb{F}_q)|$, is finite, and is upper bounded as $n{\leq}q{+}1$. Assume $n{=}q{+}1{-}t$, the value t is known as "trace of Frobenius at q"

Hasse Theorem

The number of points of an elliptic curve $n = |\mathbb{E}(\mathbb{F}_q)|{=}q{+}1{-}t$ lies between:

$$q + 1 - 2\sqrt{q} \leq n \leq q + 1 + 2\sqrt{q}$$

or, equivalently, the trace of Frobenius satisfies: $\mathtt{abs}(t) \leq 2\sqrt{q}$

Elliptic Curves: Group Structure

Theorem [M. Tsfasman; F. Voloch; H.-G. Ruck (1988)]

Working over a finite field, the group of points $\mathbb{E}(\mathbb{F}_q)$ is always either a cyclic group or the *direct product* ($\times$) of two cyclic groups. More precisely:

$$(\mathbb{E}(\mathbb{F}_q), +) \cong (\mathbb{Z}_{n_1}, \cdot) \times (\mathbb{Z}_{n_2}, \cdot) \qquad n_2 = \gcd(n_1, q-1)$$

with only few fully described exceptions

E.g., $P_1{\cong}(a, b)$, $P_2{\cong}(c, d)$; for some $a, c{\in}\mathbb{Z}_{n_1}$ and $b, d{\in}\mathbb{Z}_{n_2}$,
$P_1{+}P_2{\cong}(a, b){\times}(c, d){=}(a \cdot c, b \cdot d)$

The *direct product* between cyclic groups is aka *direct sum*: $\mathbb{Z}_{n_1}{\oplus}\mathbb{Z}_{n_2}$

Corollary

if $n{=}|\mathbb{E}(\mathbb{F}_q)|$ is equal to the product of distinct primes, then $(\mathbb{E}(\mathbb{F}_q), +)$ is cyclic

Proof.

if $n{=}p_1\,p_2$ then $p_1\,p_2{=}n_1\,n_2$, but as $n_2|n_1 \Rightarrow n_2 = 1$.
Being $\mathbb{Z}_{n_1}$ cyclic (see the statement of the Theorem) then also $\mathbb{E}(\mathbb{F}_q)$ is cyclic $\square$

Elliptic Curve Cryptography (ECC)

For any elliptic curve and any finite field, $n{=}|\mathbb{E}(\mathbb{F}_q)|$ can be computed in $O(\log_2^5 q)$ with the *Schoof–Elkies–Atkin algorithm*

We do not see the details of the point-counting algorithm, you should just remember that computing the group order is "easy"

- In practice, only curves with "nearly" prime cardinality are considered: $|\mathbb{E}(\mathbb{F}_q)|{=}c\,p$, where $p \geq 2^{160}$
- To properly work in the subgroup with prime order p, we need to compute one of its generators
  - it is sufficient to consider a generic point $Q(x_Q, y_Q)$ over the curve and test whether $[c]Q \neq \mathcal{O}$ or not
  - If the test is passed, then $P = [c]Q$ is a generator of the subgroup with prime order

Elliptic Curve Discrete Log. Problem

The additive cyclic subgroup of order p described above is similar to the multiplicative group of powers of an integer g modulo a prime p:

$$(\mathcal{O}, P, [2]P, [3]P, \ldots, [p-1]P) \Leftrightarrow (1, g, g^2, g^3, \ldots, g^{p-1})$$

ECDLP

Given a prime group $(G, +) \subseteq \mathbb{E}(\mathbb{F}_q)$, and two points $P, Q \in G$ the EC discrete log. problem asks to find the smallest integer k ($0 \leq k < |G|$) such that

$$Q = [k]P = \underbrace{P + P + \ldots + P}_{k-\text{times}}$$

to compute scalar-point multiplication $[k]P$ we employ the usual "square & multiply" strategy

- The only change is the way we denote the group operation
- In case of ECs this strategy is also called "Double & Add"

Scalar Point Multiplication

- The security of elliptic curve cryptography (ECC) is based on the hardness of the ECDLP over a group G with prime order
- The best known algorithms to solve it have fully exponential complexity $O(\log^{\frac{1}{2}} |G|)$
- The sizes of the EC group recommended by NIST are:

Symmetric Key size [bit]	size of ECC prime order group [bit]	RSA/DH/DSA (key lengths-group size) [bit]
80	160	1024
112	224	2048
128	256	3072
192	384	7689
256	512	15360

- **The NIST standardized a set of elliptic curves to be used in EC-based cryptosystems** (both over $\mathbb{F}_p$ and $\mathbb{F}_{2^m}$)
- care was put in their definition to obtain particularly fast finite field arithmetic

Elliptic Curve Cryptography (ECC)

- One of the problems with the formulae for the group laws is that at some stage they involve a division operation
- We have seen that division in finite fields (i.e., inverse computation) is an "expensive" operation that cannot be implemented as efficiently as modular multiplications or additions
- To avoid divisions in the group law formulae, the elliptic curves are considered in "projective coordinates"

Elliptic Curve Cryptography (ECC)

For mathematicians the Projective plane is the more natural context to deal with the elliptic curves properties

Let $\mathbb{K}$ be a field. The "projective plane" $\mathbb{P}^2(\mathbb{K})$ over $\mathbb{K}$ is defined as the set of triples (X, Y, Z) where $X, Y, Z \in \mathbb{K}$ are not all simultaneously zero: this is a "projective point". On these triples is defined an equivalence relation

$$(X, Y, Z) \equiv (\lambda X, \lambda Y, \lambda Z) \text{ with } \lambda \in \mathbb{K} \setminus \{0\}$$

E.g., if $\mathbb{K}=\mathbb{F}_{11}$, then the two points $(4, 3, 1)$ and $(1, 9, 3)$ are equivalent $(\lambda = 3)$

Elliptic Curve Cryptography (ECC)

Correspondence between Projective and Affine Planes

- The projective plane (X, Y, Z) is identified with the affine one (x, y) plus the adjoint of the "points at infinity"
- If a projective point (X, Y, Z) has $Z \neq 0$, then there exists a unique t-uple in its equivalence class $(x, y, 1)$ where $x=X/Z$, $y=Y/Z$ that can be put in one-to-one correspondence with the affine point (x, y)
- If a projective point has $Z=0$, it is called point at infinity
- A generic point at infinity $(X, Y, 0) \sim (1, Y/X, 0)$ can be put in one-to-one correspondence with **the direction of the straight lines having slope** $m=Y/X$
  - points at infinity can be visualized as the "horizon" of the affine plane

Elliptic Curve Cryptography (ECC)

Elliptic Curve

Let $\mathbb{K}$ be a field. An elliptic curve is defined as the set of solutions in the projective plane of a non-singular homogeneous Weierstrass equation of the form

$$\mathbb{E}(\mathbb{K}): \; Y^2Z + a_1XYZ + a_3YZ^2 = X^3 + a_2X^2Z + a_4XZ^2 + a_6Z^3$$

The points at infinity of the previous curve can be found through

$$\begin{cases} Y^2Z + a_1XYZ + a_3YZ^2 = X^3 + a_2X^2Z + a_4XZ^2 + a_6Z^3 \\ Z = 0 \end{cases} \Rightarrow$$

$$\Rightarrow \quad (\text{a single point}) \; \mathcal{O} = (0, \lambda, 0), \; \lambda \in \mathbb{K} \setminus \{0\}$$

The direction of the vertical lines parallel to the y-axis is conventionally denoted as $\mathcal{O}$=(0, 1, 0)

Elliptic Curve Cryptography (ECC)

Addition Formulae

$$\mathbb{E}(\mathbb{F}_{p^m}): \; Y^2Z = X^3 + aXZ^2 + bZ^3, \; 4a^3 + 27b^2 \neq 0, \; p > 3$$

$$P_1 = (X_1, Y_1, Z_1), \; P_2 = (X_2, Y_2, Z_2), \; P_3 = P_1 + P_2 = (X_3, Y_3, Z_3)$$

$$\begin{cases} X_3 = p_1(t - q_2) \\ 2Y_3 = r(3q_2 - 2t) - p_3(s_2 + s_1) \\ Z_3 = wp_3 \end{cases}$$

$$\begin{array}{ll} u_1 = X_1Z_2 & p_2 = p_1^2 \\ u_2 = X_2Z_1 & p_3 = p_1p_2 \\ s_1 = Y_1Z_2 & q_1 = u_1 + u_2 \\ s_2 = Y_2Z_1 & q_2 = p_2q_1 \\ w = Z_1Z_2 & r = s_2 - s_1 \\ p_1 = u_1 - u_2 & t = wr^2 \end{array}$$

Only multiplications and additions in the finite filed are employed

Elliptic Curve Cryptography (ECC)

Doubling Formulae

$$\mathbb{E}(\mathbb{F}_{p^m}): \; Y^2Z = X^3 + aXZ^2 + bZ^3, \; 4a^3 + 27b^2 \neq 0, \; p > 3$$

$$P_1 = (X_1, Y_1, Z_1), \; P_3 = [2]P_1 = P_1 + P_1 = (X_3, Y_3, Z_3)$$

$$\begin{cases} X_3 = \lambda_1(\lambda_4 - 4\lambda_5) \\ Y_3 = \lambda_2(6\lambda_5 - \lambda_4) - 2Y_1^2\lambda_6 \\ Z_3 = \lambda_1\lambda_6 \end{cases}$$

$$\begin{array}{ll} \lambda_1 = 2Z_1Y_1 & \lambda_4 = \lambda_2^2 \\ \lambda_2 = 3X_1^2 + aZ_1^2 & \lambda_5 = \lambda_1\lambda_3 \\ \lambda_3 = X_1Y_1 & \lambda_6 = \lambda_1^2 \end{array}$$

Only multiplications and additions in the finite filed are employed

Elliptic Curve Cryptography (ECC)

In practical implementations, several projective coordinate systems are employed, depending on

- the specific finite field
- the average number and the relative efficiency of each arithmetic operation (mul.s, add.s, doublings)
- how many point additions or how many point doubling a specific primitive is supposed to execute

Elliptic Curve Cryptography (ECC)

Coordinate Systems

- in the projective system each point (x, y) is represented by three coordinates (X, Y, Z) using the following relation: $x = \frac{X}{Z}$, $y = \frac{Y}{Z}$
- in the Jacobian system (also adopted for the NIST standard curves) a point (x, y) is also represented with three coordinates (X, Y, Z), but $x = \frac{X}{Z^2}$, $y = \frac{Y}{Z^3}$
- in the López–Dahab system the relation is $x = \frac{X}{Z}$, $y = \frac{Y}{Z^2}$
- in the modified Jacobian system the same relations are used but four coordinates are stored and used for calculations (X, Y, Z, aZ^4)
- in the Chudnovsky-Jacobian system five coordinates are used (X, Y, Z, Z^2, Z^3)

Elliptic Curve Diffie-Hellman (ECDH)

ECDH is a DH anonymous key agreement protocol that allows two parties, each having an elliptic curve public-private key pair, to establish a shared secret over an insecure channel

- The parties agree on the use of a publicly known group $(G, +)$, $G \subseteq \mathbb{E}(\mathbb{F}_q)$ with order $n = |G|$ and generator P
- Both parties A, B independently choose their private keys: $k_{priv,B} \in \{2, \ldots, n-1\}$, $k_{priv,A} \in \{2, \ldots, n-1\}$
- B computes $k_{pub,B} = [k_{priv,B}]P$ and sends it to A
- A computes $k_{pub,A} = [k_{priv,A}]P$ and sends it to B
- Finally, both parties compute a common point $P_{AB} = P_{BA} = [k_{priv,A} \cdot k_{priv,B}]P$

EC-ElGamal Encryption

As in the case of the DH key exchange protocol, also the ElGamal encryption algorithm can be extended to elliptic curves through choosing a group $(G, +)$, $G \subseteq \mathbb{E}(\mathbb{F}_q)$ with order $n = |G|$ and generator $P(x_P, y_P)$

- Assume a user A is willing to send an encrypted message to B, then she maps the msg into a curve point $M \in \mathbb{E}$
- User B is equipped with a
  - secret key: $0 < k_{priv,B} < n$
  - public key: $k_{pub,B} = (n, P, [k_{priv,B}]P)$
- to encrypt the msg M and to send it to B, the sender A
  - picks $r \xleftarrow{\text{Random}} \{1, \ldots, n-1\}$
  - computes $\gamma \leftarrow [r]P$ and $\delta \leftarrow M + [r]k_{pub,B}$
  - sends the ctx (γ, δ)
- After receiving the ctx (γ, δ), the receiver B:
  - computes $[k_{priv,B}]\gamma = [k_{priv,B}]([r]P) = [r]k_{pub,B}$
  - decrypts the msg as: $\delta - [k_{priv,B}]\gamma = (M + [r]k_{pub,B}) - ([r]k_{pub,B})$

EC-ElGamal Encryption

Due to the fact that the definition of a **bijective mapping** between a binary string and a point over the EC **is quite challenging to achieve**, the following variant is usually preferred:

- A sends a message m encoded in binary with the same bitsize of x_P
- User B is equipped with a
  - secret key: $0 < k_{priv,B} < n$
  - public key: $k_{pub,B} = (n, P, [k_{priv,B}]P)$
- to encrypt the msg m and to send it to B, the sender A
  - picks $r \xleftarrow{\text{Random}} \{1, \ldots, n-1\}$
  - computes $\gamma \leftarrow [r]P$ and $\delta \leftarrow \mathbf{m} \oplus \texttt{x-coord}([\mathbf{r}]\mathbf{k_{pub,B}})$
  - sends the ctx (γ, δ) ... the $\oplus$ symbol denotes the bitwise xor operator
- After receiving the ctx (γ, δ), the receiver B:
  - computes $[k_{priv,B}]\gamma = [k_{priv,B}]([r]P) = [r]k_{pub,B}$
  - decrypts the msg as: $\delta \oplus \texttt{x-coord}([k_{priv,B}]\gamma) = m$

EC-DSA

Choose a group $(G, +)$, $G \subseteq \mathbb{E}(\mathbb{F}_q)$ with order $n = |G|$ and generator $P(x_P, y_P)$
The private/public key-pair is $k_{priv} = (s), s \in \mathbb{Z}_n$ and $k_{pub} = (n, P, [s]P)$.
In order to sign the msg $m \in \{0,1\}^*$, the sender A:

1. picks $r \overset{\text{Random}}{\leftarrow} \{1, \ldots, n-1\}$, $\gcd(r, n) = 1$
2. computes $[r]P = (x_1, y_1)$, $k \leftarrow x_1 \ mod \ n$. if $k = 0$ then go to step 1
3. computes $r^{-1} \ mod \ n$ and $e \leftarrow SHA\text{-}1(m)$
4. computes $z \leftarrow r^{-1}(e + sk) \ mod \ n$. if $z = 0$ then go to step 1
5. The signature of m is (k, z)

To verify the signature, the receiver B:

1. Checks if $k, z \in \{1, \ldots, n-1\}$
2. computes $e \leftarrow SHA\text{-}1(m)$ and $w \leftarrow z^{-1} \ mod \ n$.
3. computes $u_1 \leftarrow ew \ mod \ n$ e $u_2 = kw \ mod \ n$.
4. computes $X \leftarrow u_1 P + u_2 k_{pub,A} = (x_1, y_1)$.
   If $X = \mathcal{O}$ then the msg is rejected, otherwise accept the signature iif $x_1 \ mod \ n = k$

When the signature (k, z) is correct, we have that $z = r^{-1}(e + sk) \ mod \ n$, thus:

$$u_1 P + u_2 k_{pub,A} = ez^{-1}P + kz^{-1}(sP) = (z^{-1}(e + sk))P = rP = (x_1, y_1) \Rightarrow k = x_1 \ mod \ n$$