Egzaminas Sausio 6 d., V-tad., 17:30-19:30, 506 a.

Elliptic-curve cryptography (**ECC**) is an approach to public-key cryptography based on the algebraic structure of elliptic curves over finite fields.
ECC requires smaller keys compared to non-ECC cryptography to provide equivalent security.

2^{256} ↓ RSA ~2^{3000}

$\mathcal{I}_{11} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$ + mod 11, - mod 11, * mod 11, / mod 11

$1 - 1 = 1 + (-1) = 0$

$1 + (-1) \bmod 11 = (1 + 10) \bmod 11 = 0$

$-1 \bmod 11 \equiv 10 \in \mathcal{I}_{11}$

$1 + 10 \bmod 11 = 11 \bmod 11 = 0$

11 + 11 = 1, 11, 0

Elliptic curves are applicable for key agreement, digital signatures, pseudo-random generators and other tasks.
Indirectly, they can be used for encryption by combining the key agreement with a symmetric encryption scheme.

Elliptic Curve Digital Signature Algorithm - Bitcoin Wiki **(ECDSA)**
https://en.bitcoin.it/wiki/Elliptic_Curve_Digital_Signature_AlgorithmFeb 10, 2015
Elliptic Curve Digital Signature Algorithm or **ECDSA** is a cryptographic algorithm used by Bitcoin to ensure that funds can only be spent by their owner.
https://en.wikipedia.org/wiki/Elliptic-curve_cryptography

Finite Field denoted by $\boldsymbol{F_p}$ (or rarely $\boldsymbol{Z_p}$) when $\boldsymbol{p}$ is prime:
$\boldsymbol{Z_p}=\{0, 1, 2, 3, \ldots, \boldsymbol{p}\text{-}1\}$; $+_{\mathbf{mod}\ p}$, $-_{\mathbf{mod}\ p}$, $\bullet_{\mathbf{mod}\ p}$, $:_{\mathbf{mod}\ p}$.
Cyclic Group:
$\boldsymbol{Z_p}^* = \{1, 2, 3, \ldots, \boldsymbol{p}\text{-}1\}$; $\bullet_{\mathbf{mod}\ p}$, $:_{\mathbf{mod}\ p}$.

R

Python 3.9.1

$\boldsymbol{p}$=11, $\boldsymbol{g}$=2.

x	0	1	2	3	4	5	6	7	8	9	~~10~~
2^x mod p	1	2	4	8	5	10	9	7	3	6	~~1~~

This part can be omited

p = 11 p = 11

Multiplicative Group Z_p^*	Additive Group Z_{p-1}^+
$\boldsymbol{Z_p}^*=\{1, 2, 3, \ldots, \boldsymbol{p}\text{-}1\}$	$\boldsymbol{Z_{p-1}}^+=\{0, 1, 2, 3, \ldots, \boldsymbol{p}\text{-}2\}$

Multiplicative Group Z_p^*	Additive Group Z_{p-1}^+
$Z_p^*=\{1, 2, 3, \ldots, p-1\}$	$Z_{p-1}^+=\{0, 1, 2, 3, \ldots, p-2\}$
Operation: multiplication mod p	Operation: addition mod $(p-1)$
Neutral element is **1**.	Neutral element is **0**.
Generator g: $Z_p^*=\{ g^i; i=0,1,2, \ldots, p-1\}$ Two criterions to find g when p is strong prime. $g^n=1$ mod p and $g^n \neq 1$ mod p if $n<p$.	Generator g: $Z_{p-1}^+=\{i \bullet g; i=0,1, 2,\ldots,p-2\}$ E.g. $g=1$. $(p-1) \bullet g=0$ mod $(p-1)$ and $n \bullet g \neq 0$ mod $(p-1)$ if $n<p-2$.
Modular exponent: $t=g^k$ mod p $t= g \bullet g \bullet g \bullet \ldots \bullet g$ mod p; k–times.	Modular multiplication: $t=k \bullet g$ mod $p-1$ $t= g+g+g+ \ldots +g$ mod $p-1$; k–times.

p = 11, p-1 = 10

$\bullet_{\text{mod } p}$

$Z_{11}^*=\{1, 2, \ldots, 10\}$

$|Z_{11}^*|=10$, $g=2$.

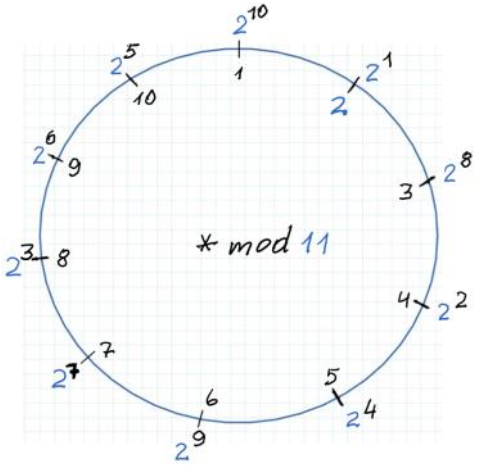
p = 11, p-1 = 10

$+_{\text{mod } (p-1)}$

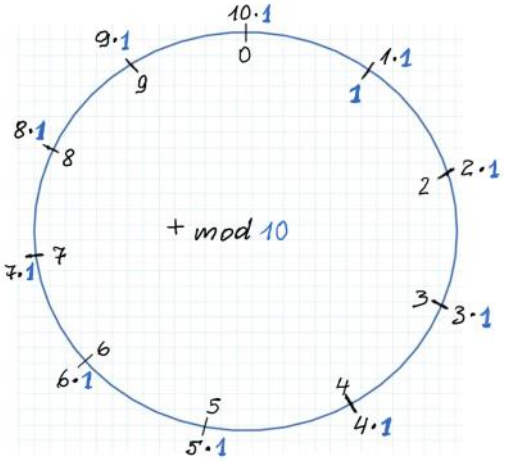
$Z_{10}^+=\{0, 1, 2, \ldots, 9\}$

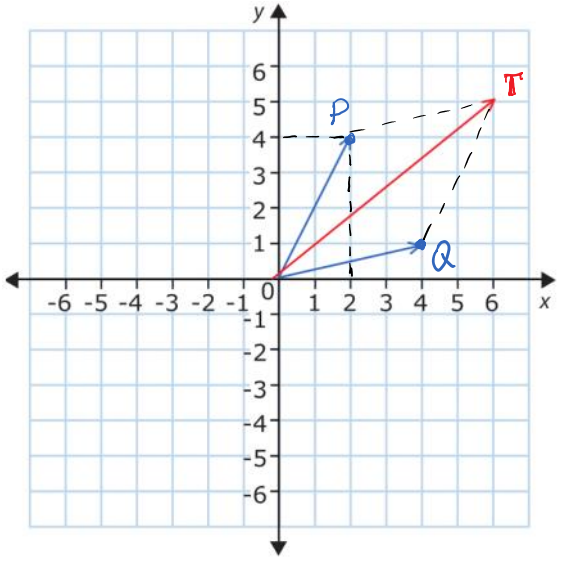
$|Z_{10}^+|=10$; $g=1$.

x		2^x
		mod 11
0	→	1
1	→	2
2	→	4
3	→	8
4	→	5
5	→	10
6	→	9
7	→	7
8	→	3
9	→	6
~~10~~	→	1



+		x•1
		mod 10
0	→	1
1	→	2
2	→	3
3	→	4
4	→	5
5	→	6
6	→	7
7	→	8
8	→	9
9	→	0
10	→	1





$P(x_P, y_P) = (2,4)$ $Q(x_Q, y_Q) = (4,1)$ $\Rightarrow P+Q=(2+4, 4+1)$ $T = P+Q = (6,5)$

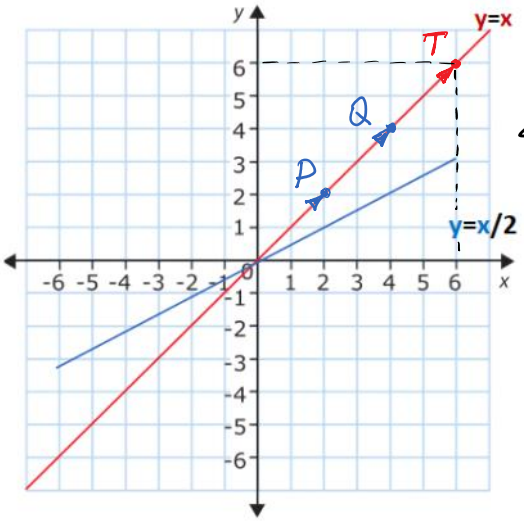
$T = P(x_P, y_P) \boxplus Q(x_Q, y_Q) =$

$= T(x_P + x_Q, y_P + y_Q) = T(x_T, y_T)$

$x_T = x_P + x_Q$

$y_T = y_P + y_Q$

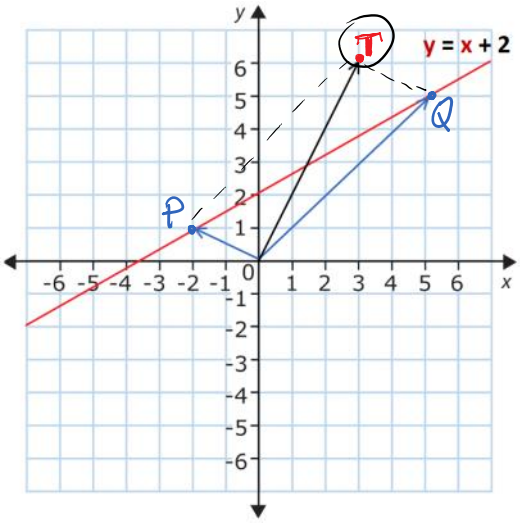
$T_2 = P+P = 2P =$

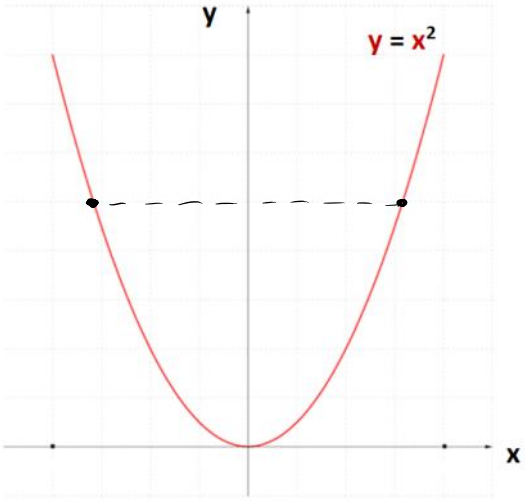


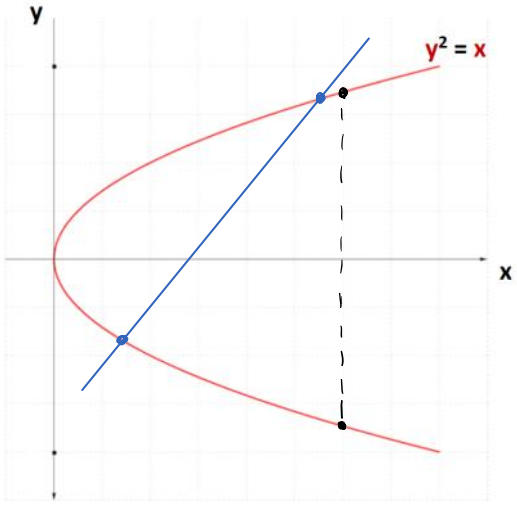
$x_T = 2+4 = 6$

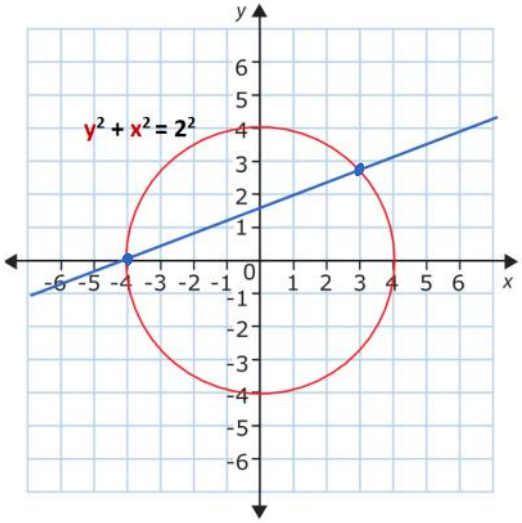
$y_T = 2+4 = 6$

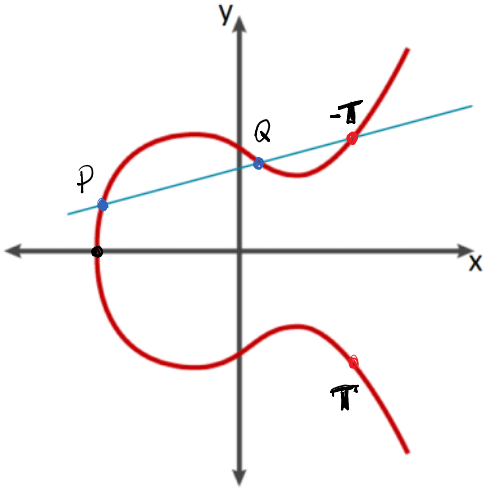
$|T| = \sqrt{6^2+6^2}$











1. symmetry

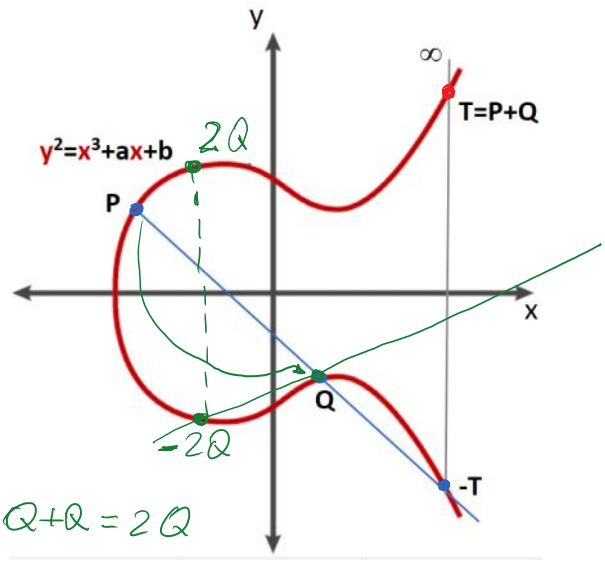
2.

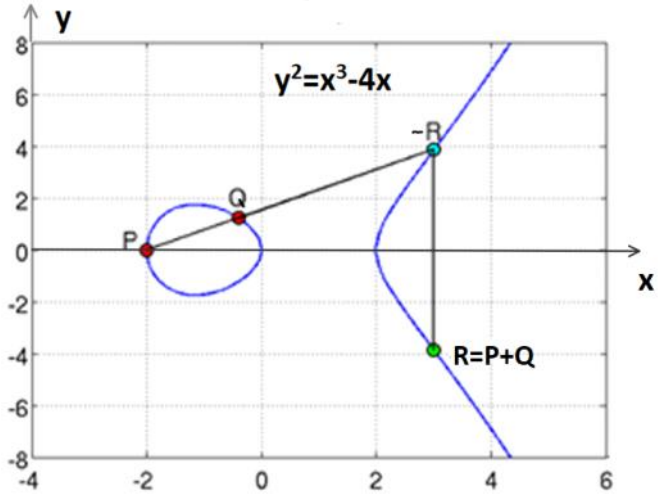
Elliptic curve ha a property that if line crosses two points, then there is a third crossing point in the curve.

Points in the plane or plane curve we denote by the capital letters, e.g. **A**, **G**, **P**, **Q**, etc.
Numbers-scalars we denote by the lowercase letters, e.g., **a**, **g**, **x**, **y**, **z**, etc.

Addition of points P and Q in EC: **P + Q = T**

$P(x_P,y_P) + Q(x_Q,y_Q) = T(x_T,y_T)$





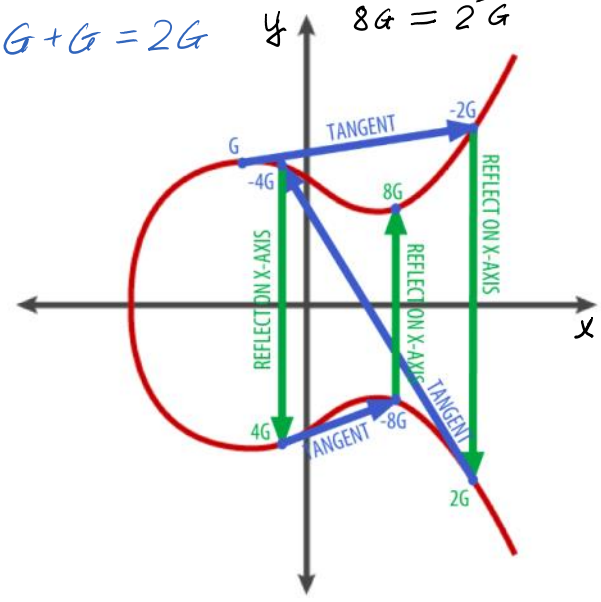
$5-5 \bmod 10 = 0$ $\quad T-T = \text{"}0\text{"} \rightarrow T+(-T) = \text{"}0\text{"} \equiv \infty$

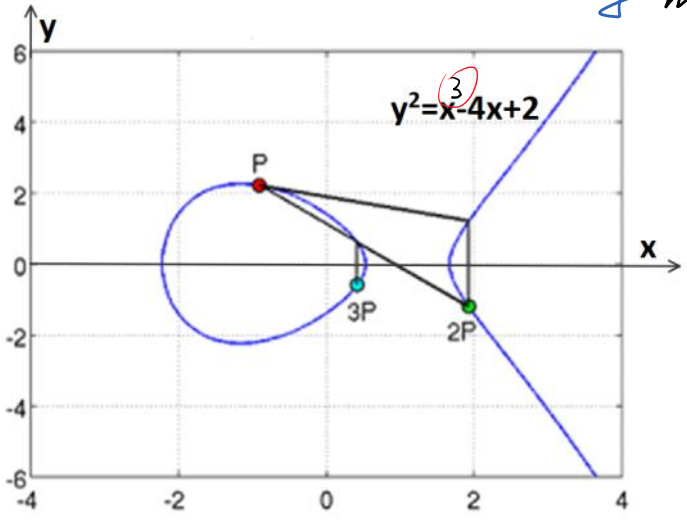
$7+0 \bmod 10 = 7$ $\quad T+\infty = T$

When z is large, $z \sim 2^{256} \rightarrow |z| = 256$ bits:

Doubling of points allows effectively compute point $A = zG$ $\quad a = g^x \bmod p$

$G+G=2G$ $\quad 8G = 2^3 G$





ECDSA animacija

Signing and Verifying Ethereum Signatures – Yos Riady · Software Craftsman

https://medium.com/coinmonks/elliptic-curve-cryptography-6de8fc748b8b

For current cryptographic purposes, an *elliptic curve* is a plane curve over a finite field
Z_p={0, 1, 2, 3, …, **p**-1}, (rather than the real numbers)
which consists of the points satisfying the equation in Z_p

$$y^2=x^3+ax+b \mod p$$

along with a distinguished point at infinity, denoted by 0 (∞).
Finite field is an algebraic structure, where 4 algebraic operations: $+_{\mod p}$, $-_{\mod p}$, $\times_{\mod p}$, $:_{\mod p}$ are defined except the division by 0 excluded.

secp 256k1

Elliptic Curve Group (ECG)
Number of points **n** of Elliptic Curve with coordinates (***x***, ***y***) is an order of ECG, (**n** = ***p***).
Addition operation ⊞ of points in ECG: let points $P(x_P,y_P)$ and $Q(x_Q,y_Q)$ are in EC with coordinates (x_P,y_P) and (x_Q,y_Q) then $P \boxplus Q = T$ with coordinates (x_T,y_T) in EC.
Neutral element is group zero 0 at the infinity of [XOY] plane, denoted by ∞.
Multiplication of any EC point ***G*** by scalar ***z***: $T=z*G$; $T=G \boxplus G \boxplus G \boxplus ... \boxplus G$; ***z***–times.
Generator – Base Point ***G***: ECG={ $i*G$; ***i***=1,2,…,**n**}; $n*G=0$ and $q*G \neq 0$ if **q**<**n**.

ElGamal Cryptosystem (CS)	**Elliptic Curve Cryptosystem (CS)**
PP=(strongprime=***p***, generator=***g***) ***p*=255996887**; ***g*=22**;	**PP**=(EC=**secp256k1**, BasePoint=***G***, prime=***p***) |EC Group order|=**N** points in EC: **N=*p*;**
PrK=***x*** >> **x**=randi(***p***-1) % or >> **x**=randi(**p**-1)	**PrK** $_{ECC}$**=z**, |EC Group order|=**N**=***p*** points in EC >> **z**=randi(**p**-1) ; z<p ; |p| = 256 bits
PuK=$A=g^x \mod p$	**PuK**$_{ECC}$=A=**z***G
Alice ***A***: **x=1975596**; ***a*=210649132**;	**Alice** ***A***: **z=…..**; $A=(a_x, a_y)$;
	|z| = 256 bits |A| = 512 bits

Generator

Elliptic Curve Digital Signature Algorithm - Bitcoin Wiki
https://en.bitcoin.it/wiki/Elliptic_Curve_Digital_Signature_AlgorithmFeb 10, 2015
Elliptic Curve Digital Signature Algorithm or **ECDSA** is a cryptographic algorithm used by Bitcoin to ensure that funds can only be spent by their owner.
https://en.wikipedia.org/wiki/Elliptic-curve_cryptography

ECDSA standards → Standards for Efficient Cryptography Group (SEC)
http://www.secg.org/
Bitcoin follows the **secp256k1** standard.
Public Parameters: **PP**=(EC=**secp256k1**; BasePoint=***G***; finite fielf F_p of characteristic ***p***, ***p*** is prime;
secp256k1: $y^2=x^3+a \cdot x+b \mod p$
Addition operations of ***points*** in the Elliptic Curve (EC).
To perform these operations it is required to perform an arithmetic operations with
$x, y, a, b \in Z_p$.
F_p={0, 1, 2, 3, …, ***p***-1}; $+_{\mod p}$, $-_{\mod p}$, $\bullet_{\mod p}$, $:_{\mod p}$.

BasePoint ***G*** is a generator of *additive* EC Group of points **secp256k1**.
Then number of points in EC Group is |EC Group|=**p**, where |**p**|=256.

Private Key of EC Cryptosystem (ECC) is $\mathbf{PrK}_{\mathbf{ECC}}=\mathbf{z}$, where **z** is secret integer generated at random, i.e. **z** ← randi.
Public Key of ECC is $\mathbf{PuK}_{\mathbf{ECC}}=A=\mathbf{z}*G$,
where * means **z**-times addition of points ***G*** in EC, i.e. multiplication of ***G*** by integer **z**.

Property 1: $(u+v)*P = u*P \boxplus v*P$ replacement to --> $\underline{(u+v)P = uP+vP}$
Property 2: $(u)*(P \boxplus Q) = u*P \boxplus u*Q$ replacement to --> $u(P+Q) = uP+uQ$

Important identity used e.g. in Ring Signature:
$(t-zc)*G+c*A = t*G-zc*G+c*A = t*G-c(z*G)+c*A = t*G-c*A+c*A = tG$.

The generation of domain parameters is not usually done by each participant because this involves computing the number of points on a curve which is time-consuming and troublesome to implement.
As a result, several standard bodies published domain parameters of elliptic curves for several common field sizes.
Such domain parameters are commonly known as "standard curves" or "named curves"; a named curve can be referenced either by name or by the unique object identifier defined in the standard documents:

- NIST, Recommended Elliptic Curves for Government Use
- SECG, SEC 2: Recommended Elliptic Curve Domain Parameters
- ECC Brainpool (RFC 5639), ECC Brainpool Standard Curves and Curve Generation

SECG test vectors are also available.[9]
NIST has approved many SECG curves, so there is a significant overlap between the specifications published by NIST and SECG.
EC domain parameters may be either specified by value or by name.
From <https://en.wikipedia.org/wiki/Elliptic-curve_cryptography>

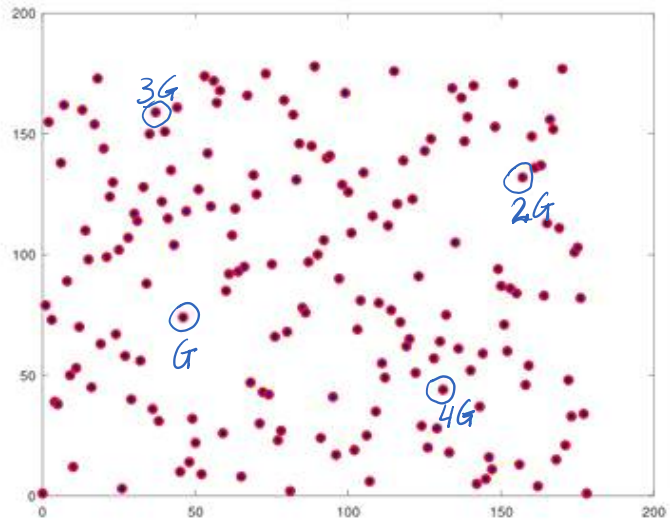
Suppose Alice wants to send a signed message to Bob. Initially, they must agree on the domain parameters which represents the Public Parameters **PP** = (**EC**, **G**, **p**):
EC - is Elliptic Curve type;
G - is a base point (generator) of EC;
p - is a number of points of Elliptic Curve, e.g. **n**=***p*** and ***p*** is prime, where operations are performed mod **p**.

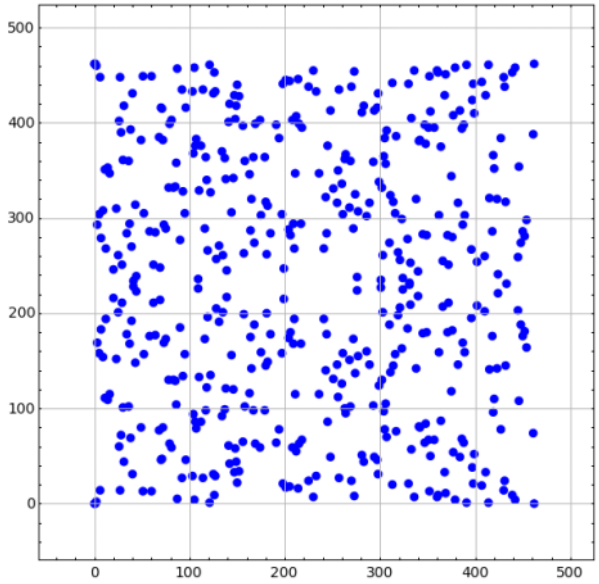
PP	
EC	the elliptic curve field and equation used $\mathbf{y^2=x^3+a \cdot x+b \ \ mod}\ p$
G	elliptic curve base point, a generator of the elliptic curve with large prime order ***p***
p	Prime integer order of ***G***, means that **p***G=0

SHA-256

Public Parameters: **PP** = (***EC***, ***G***, ***p***), ***G***=(***x*****$_G$**, ***y*****$_G$**)

https://doc.sagemath.org/html/en/thematic_tutorials/explicit_methods_in_number_theory/elliptic_curves.html

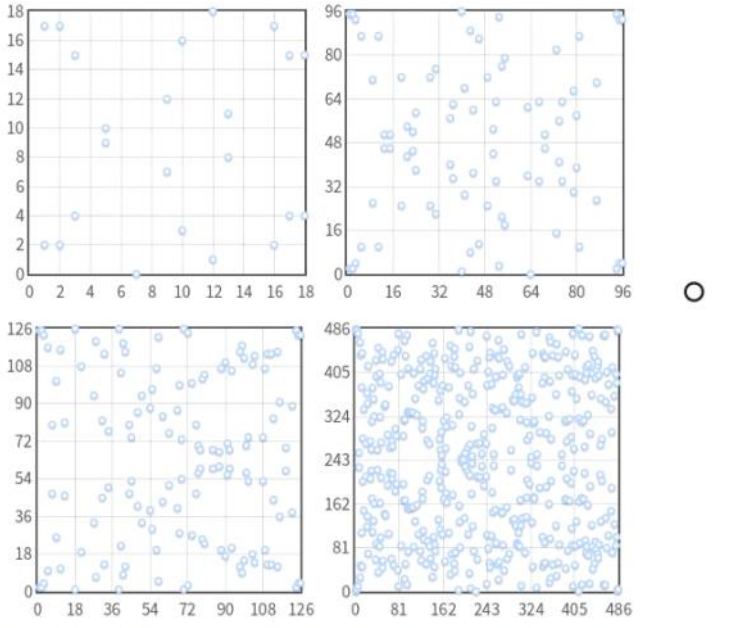




https://andrea.corbellini.name/2015/05/23/elliptic-curve-cryptography-finite-fields-and-discrete-logarithms/

The curve $y^2 \equiv x^3 - 7x + 10 \pmod{p}$ with **p**=19,97,127,487. Note that, for every x, there are at most two points. Also note the symmetry about y=p/2.

From <https://andrea.corbellini.name/2015/05/23/elliptic-curve-cryptography-finite-fields-and-discrete-logarithms/>


The curve $y^2 = x^3 - 7x + 10 \pmod{p}$ with $p = 19, 97, 127, 487$. Note that, for every x, there are at most two points. Also note the symmetry about $y = p/2$.

http://www.secg.org/sec2-v2.pdf

SEC 2: Recommended Elliptic Curve Domain Parameters
Certicom Research Contact: Daniel R. L. Brown (dbrown@certicom.com)
January 27, 2010 Version 2.0

Recommended standard **EC** for Bitcoin and other cryptocurrencies is **secp256k1**

The other standard **EC** is **secp256r1:** $y^2=x^3+a\cdot x+b \mod p$.

Elliptic curve - EC: domain parameters over $F_p = Z_p=\{0, 1, 2, 3, \ldots, p-1\}$ are specified by the tuple $T = (p,a,b,G)$. EC equation is $y^2=x^3+a\cdot x+b \mod p$.

Fnite field F_p is defined by:

p = FFFFFFFF 00000001 00000000 00000000 00000000 FFFFFFFF FFFFFFFF FFFFFFFF ==

$== 2^{224}(2^{32}-1) + 2^{192} + 2^{96} - 1$ // p is prime

The elliptic curve **EC**: $y^2 = x^3 + ax + b \mod p$ over F_p is defined by:

a = FFFFFFFF 00000001 00000000 00000000 00000000 FFFFFFFF FFFFFFFF FFFFFFFC

b = 5AC635D8 AA3A93E7 B3EBBD55 769886BC 651D06B0 CC53B0F6 3BCE3C3E 27D2604B

EC was chosen verifiably at random as specified in ANSI X9.62 [X9.62] from the seed:

S = C49D3608 86E70493 6A6678E1 139D26B7 819F7E90

The base point G in compressed form is:

G = 03

6B17D1F2 E12C4247 F8BCE6E5 63A440F2 77037D81 2DEB33A0 F4A13945 D898C296

G in uncompressed form is:

G = 04

6B17D1F2 E12C4247 F8BCE6E5 63A440F2 77037D81 2DEB33A0 F4A13945 D98C296 x_G

4FE342E2 FE1A7F9B 8EE7EB4A 7C0F9E16 2BCE3357 6B315ECE CBB64068 37BF51F5 y_G

$G = (x_G, y_G)$ $|x_G|$=256 bits, $|y_G|$=256 bits --> $|G|$=512 bits.

The order $N$of G is:

N= FFFFFFFF 00000000 FFFFFFFF FFFFFFFF BCE6FAAD A7179E84 F3B9CAC2 FC632551

$|N| = 8\cdot8\cdot4 = 256$ bits $n*G = 0$ // 0 - is a neutral element - point at infinity.

If p is prime number and is equal to the number of points of EC, then $p = N$.

It is a case if **EC** is **secp256k1**.

The cofactor is: h = 01

What is an elliptic curve cofactor? - Cryptography Stack Exchange

In cryptography, an elliptic curve is a group which has a given size n. We normally work in a subgroup of prime order r, where r divides n. The "cofactor" is h=n/r. For every point P on the curve, the point hP is either the "point at infinity", or it has order r; i.e., when taking a point, multiplying it by the cofactor necessarily yields a point in the subgroup of prime order r.

The cofactor matters inasmuch as it is not equal to 1:

- When the cofactor is 1, then the subgroup is the whole curve. Any non-zero point is a generator. Any incoming point (x,y) that fulfills the curve equation is part of the subgroup. Everything is fine.
- When the cofactor is *not* 1, then the subgroup of prime order is a strict subset of the curve. When considering a point, verifying that the curve coordi nates match the curve equation is not sufficient to guarantee that the point is on the appropriate subgroup. Moreover, there will be points w hose order is not a multiple of r. This is what happens, for instance, with Curve25519, which has a cofactor of 8. Such curves require some extra care in the protocol that uses them. For instance, when doing a Diffie-Hellman key exchange over Curve25519, the Diffie-Hellman private keys must be chosen as multiples of 8 (which is expressed as: "set the three least significant bits to zero"); this ensures that the points will be in the proper subgroup.

$\mathbf{PrK}_{ECC}=z < N < 2^{256}$; $\mathbf{PuK}_{ECC}=A=(a_x, a_y)$;

$|\mathbf{PrK}_{ECC}=z|$=256 bits; $|\mathbf{PuK}_{ECC}=A|$=512 bits.

Doubling points in EC

$A=11*G$

$11= 1011_2 = 1\cdot2^3 + 0\cdot2^2 + 1\cdot2^1 + 1\cdot2^0 = 8 + 2 + 1 = 11.$

$11= 1011_2 = 2\cdot2\cdot2 + 0\cdot2\cdot2 + 2 + 1 = 2\cdot2\cdot2 + 2 + 1$ // $*G$

$11 = 1011_2 = 1\cdot2^3 + 0\cdot2^2 + 1\cdot2^1 + 1\cdot2^0 = 8 + 2 + 1 = 11.$

$11 = 1011_2 = 2\cdot2\cdot2 + 0\cdot2\cdot2 + 2 + 1 = 2\cdot2\cdot2 + 2 + 1$ // *G

A= 2*(2*(2*G)) ⊞ 0*G ⊞ 2*G ⊞ 1*G

A= (8*G) ⊞ 2*G ⊞ G.

Till this place

Because this curve is defined over a finite field of prime order instead of over the real numbers, it looks like a pattern of dots scattered in two dimensions, which makes it difficult to visualize. However, the math is identical to that of an elliptic curve over real numbers. As an example, Elliptic curve cryptography: visualizing an elliptic curve over F(p), with p=17 shows the same elliptic curve over a much smaller finite field of prime order 17, showing a pattern of dots on a grid. The secp256k1 bitcoin elliptic curve can be thought of as a much more complex pattern of dots on a unfathomably large grid.

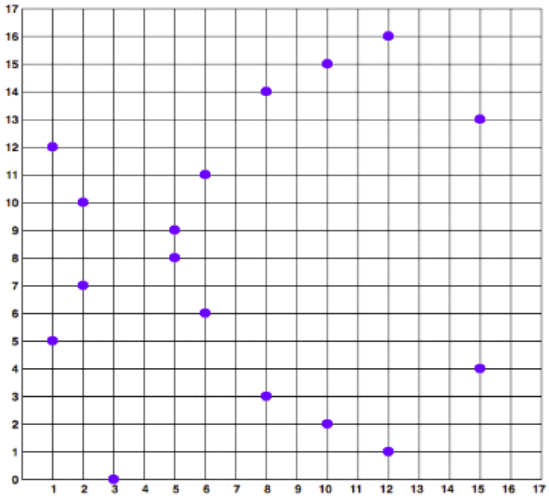


Figure 3. Elliptic curve cryptography: visualizing an elliptic curve over F(p), with p=17

$p = 17 \rightarrow Z_{17} = \{0, 1, 2, \ldots, 16\}$

$+, -, \cdot, :$ mod 17

$a, b \in Z_{17}$

N – number of point of EC

$N = p$