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.
For example, to achieve the same security ensured by ECC having private key of 256 bit length, it is required to use 3000 bit private key length for RSA cryptosystem and others.

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, Ethereum and other blockchain methods to ensure that funds can only be spent by their owner.
https://en.wikipedia.org/wiki/Elliptic-curve_cryptography

Finite Field denoted by F_p (or rarely Z_p), when: p is prime.

$F_p=\{0, 1, 2, 3, \dots, p-1\}$; $+_{\text{mod } p}$, $-_{\text{mod } p}$, $\bullet_{\text{mod } p}$, $:_{\text{mod } p}$ (except division by 0).
Cyclic Group: $Z_p^* = \{1, 2, 3, \dots, p-1\}$; $\bullet_{\text{mod } p}$, $:_{\text{mod } p}$.

p=11
$x a = \in$

For example, if p=11, then one of the generetors is g=2.

The main function used in cryptography was Discrete Exponent Function - DEF:
$\text{DEF}(x) = g^x \bmod p = a$.

x	0	1	2	3	4	5	6	7	8	9	~~10~~
$2^x \bmod p$	1	2	4	8	5	10	9	7	3	6	~~1~~

Animation

Discrete Exponent Function - $\text{DEF}_g(x)=g^x \bmod p$
x is in $Z_{p-1}= Z_{10} = \{0, 1, 2, \dots, 9\}$; mod $(p-1)$
$\text{DEF}(x)$ is in $Z_p^* = Z_{11}^* = \{1, 2, 3, \dots, 10\}$; mod p
DEF: $Z_{p-1} \rightarrow Z_p^*$.
Fermat theorem: if p is prime, then for any z: $z^{p-1}=1 \bmod p$.
If g is a generator in Z_p^* then DEF is 1-to-1 mapping.

$x \in Z_{10}$				$a \in Z_{11}^*$
0				1
1				2
2				4
3				8
4				5
5				10
6				9
7				7
8				3
9				6

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

p = 11, p-1 = 10

•mod p

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

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

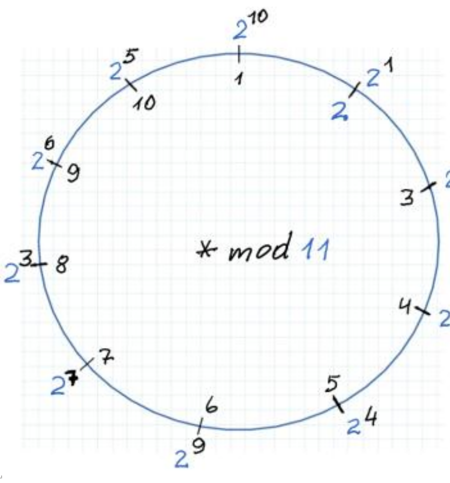
p = 11, p-1 = 10

+mod (p-1)

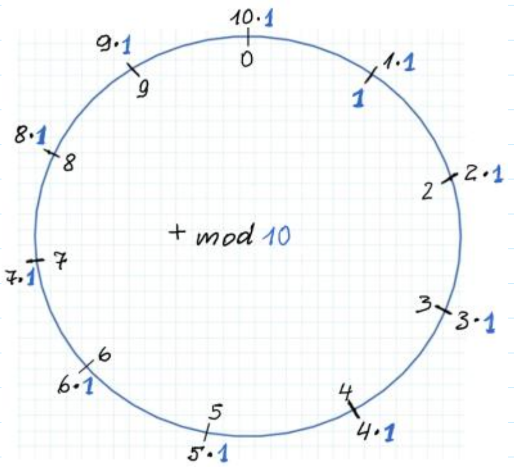
$Z_{10}^+=\{0, 1, 2, \dots, 9\}$

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

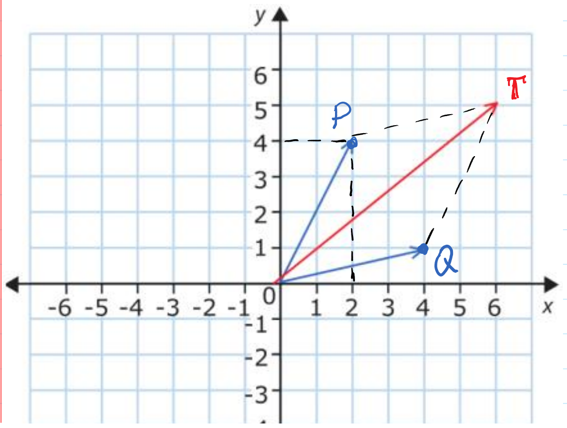
x		2^x
		mod 11
0	→	1
1	→	2
2	→	4
3	→	8
4	→	5
5	→	10
6	→	9
7	→	7
8	→	3
9	→	6
~~10~~	→	1



x		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



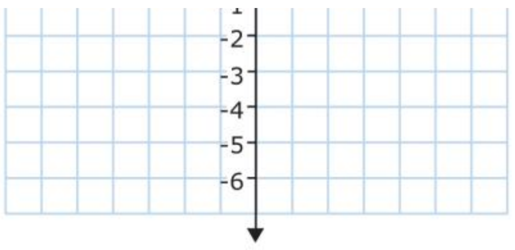
Coordinate systems XOY in subsequent examples are defined in the plane of real numbers.



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

$$T = P(x_P, y_P) \overset{\boxplus}{+} Q(x_Q, y_Q) =$$
$$= T(x_P + x_Q,\ y_P + y_Q) = T(x_T, y_T)$$

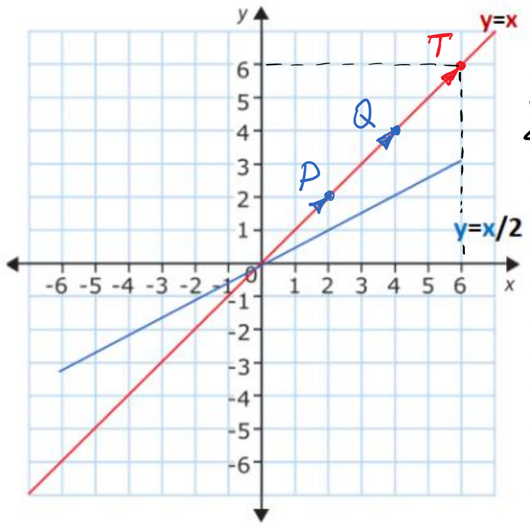
$$x_T = x_P + x_Q$$
$$y_T = y_P + y_Q$$



$x_T = x_P + x_Q$

$y_T = y_P + y_Q$

$T_2 = P + P = 2P =$

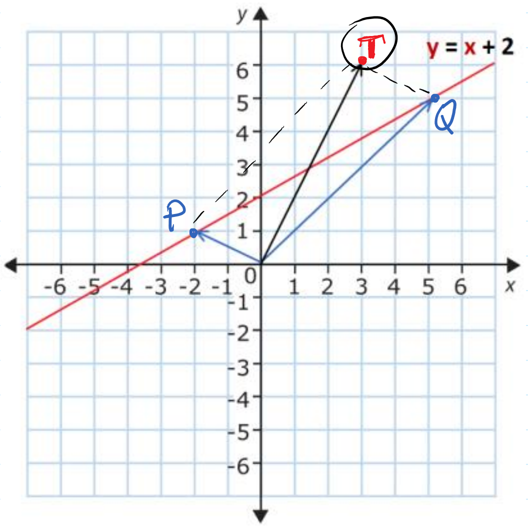


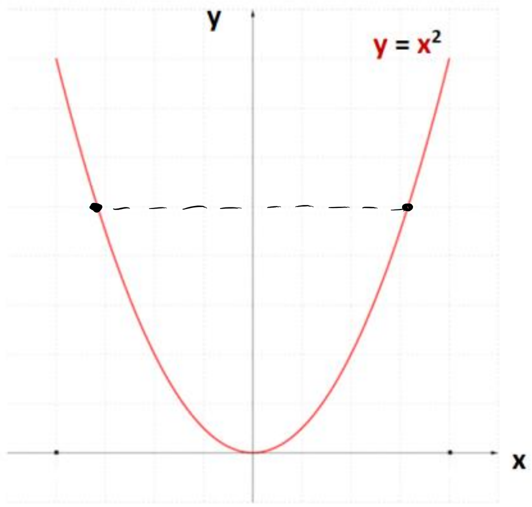
$x_T = 2 + 4 = 6$

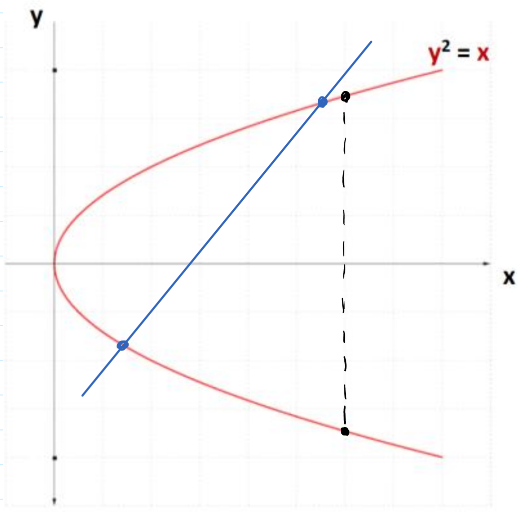
$y_T = 2 + 4 = 6$

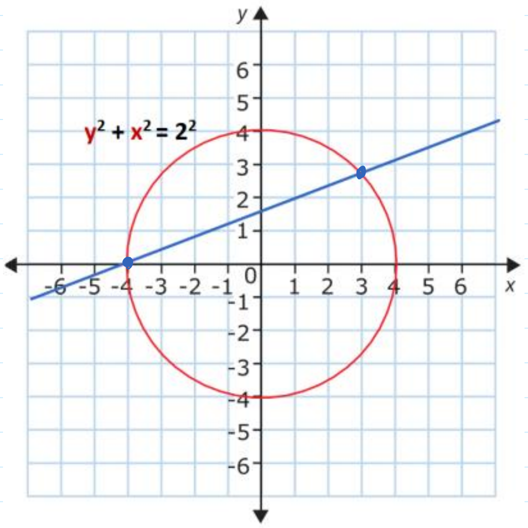
$|T| =$

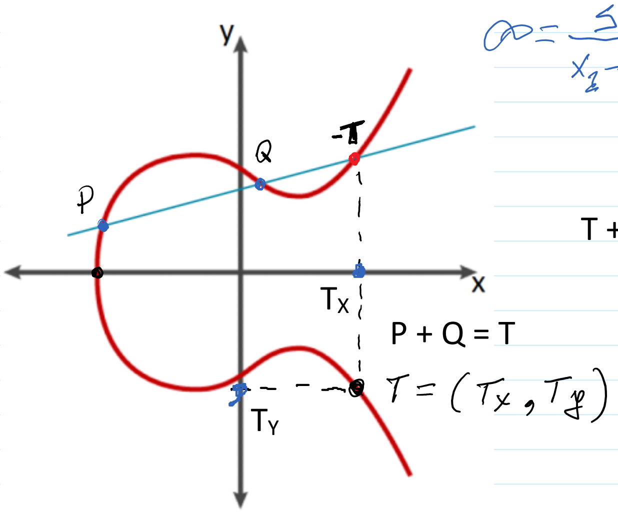
$= \sqrt{6^2 + 6^2}$











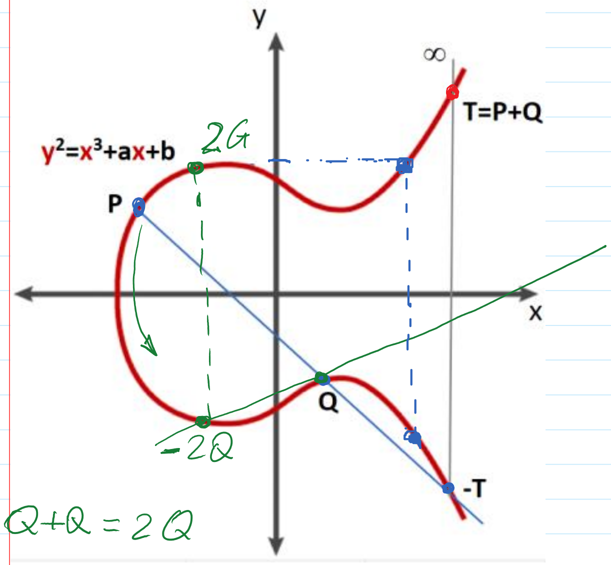
$\infty = \frac{s}{x_2 - x_1}$

T + (-T) = **0** ???

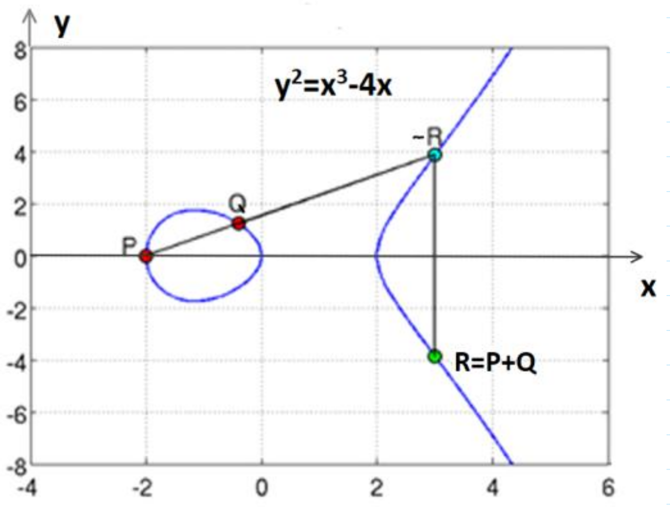
Elliptic curve ha a property that if line crosses two points, then there is a third crossing point in the curve.

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



$Q + Q = 2Q$



$5 - 5 \mod 10 = 0$ $\quad T - T = „0" \longrightarrow T + (-T) = „0" \equiv \infty$

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

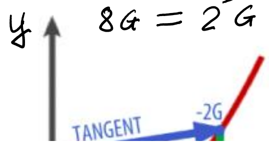
When z is large, $z \sim 2^{256} \longrightarrow |z| = 256$ bits :

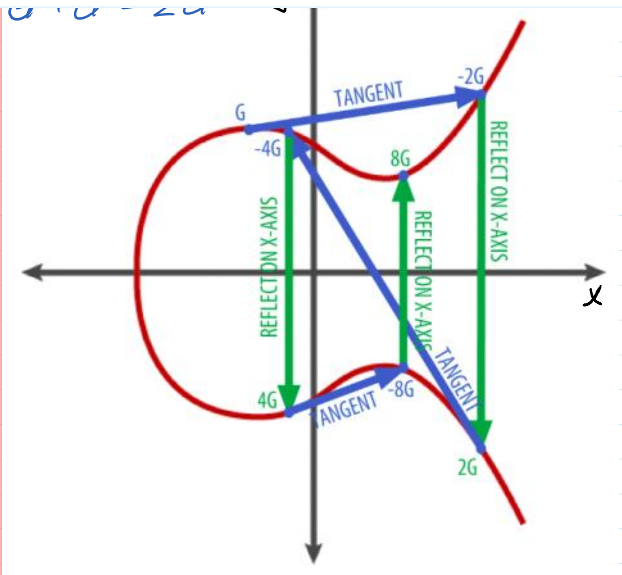
Doubling of points allows effectively compute point $A = zG$

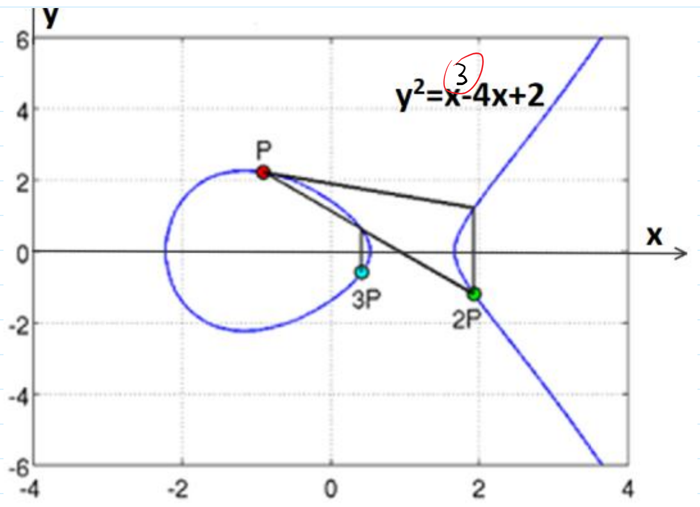
$a = g^x \mod p$

$G + G = 2G$

$8G = 2^3 G$







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
F_p={0, 1, 2, 3, …, p-1}, (rather than the real numbers) p-is prime.
Which consists of the points satisfying the equation over F_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: $+_{\text{mod } p}$, $-_{\text{mod } p}$, $´_{\text{mod } p}$, $:_{\text{mod } p}$
are defined except the division by 0 excluded.

Elliptic Curve Group (ECG)
Number of points **N** of Elliptic Curve with coordinates (***x***, ***y***) is an order of ECG.
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 $P ⊞ Q = T$ with coordinates (x_T,y_T) in EC.
Neutral element is group zero 0 at the infinity (∞) of [XOY] plane.
Multiplication of any EC point G by scalar z: $T=z*G$; $T=G ⊞ G ⊞ G ⊞ ...⊞ G$; z–times.
Generator–Base Point G: ECG={ $i*G$; i=1,2,…,N}; $N*G$=0 and $q*G≠0$ if $q<N$.

ElGamal Cryptosystem (CS)	**Elliptic Curve Cryptosystem (CS)**
PP=(strongprime p, generator g); p=255996887; g=22;	**PP**=(EC **secp256k**; BasePoint-Generator G; prime p; param. a, b); Parameters a, b defines EC equation $y^2=x^3+ax+b \mod p$ over F_p.
PrK=x; >> **x**=randi(p-1).	**PrK** $_{\text{ECC}}$=**z**; >> **z**=randi(p-1).
PuK=a=g^x mod p.	**PuK**$_{\text{ECC}}$=A=**z**$*G$.
Alice A: **x**=**1975596**; a=**210649132**;	**Alice** A: **z**=**…..**; $A=(x_A, y_A)$;

Alice A: x=1975596; a=210649132; **Alice** A: z=.....; $A=(x_A, y_A)$;

Let us consider abstract EC defined in XOY and expressed by the equation:

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

EC points are computed by choosing coordinate x and computing coordinate y^2.
To compute coordinate y it is needed to extract root square of y^2.

$$y = \pm\sqrt{y^2} \bmod p.$$

Notice that from y^2 we obtain 2 points in EC, namely y and $-y$ no matter computations are performed with integers **mod** p or with real numbers.
Notice also that since EC is symmetric with respect to x-axis, the points y and $-y$ are symmetric in EC.
Since all arithmetic operations are computed **mod** p then according to the definition of negative points in F_p points y and $-y$ must satisfy the condition

$$y + (-y) = 0 \bmod p.$$

Then evidently

$$y^2 = (-y)^2 \bmod p.$$

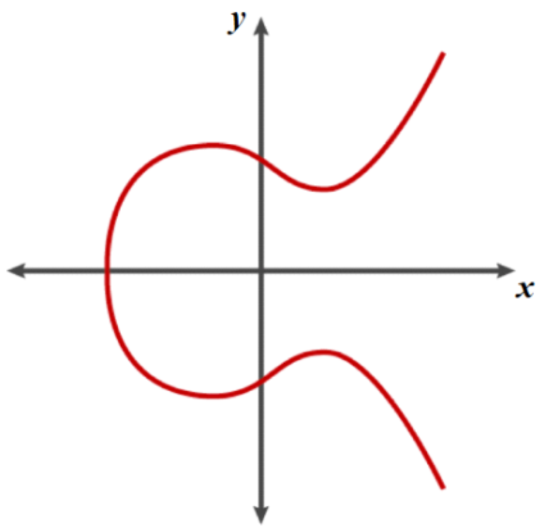
For example:
-2 mod 11 = 9
2^2 mod 11 = 4 & 9^2 mod 11 = 4

```
>> mod(9^2,11)
ans = 4
```

```
>> m2=mod(-2,11)
m2 = 9
```

```
>> mod_exp(2,2,11)
ans = 4
>> mod_exp(9,2,11)
ans = 4
```

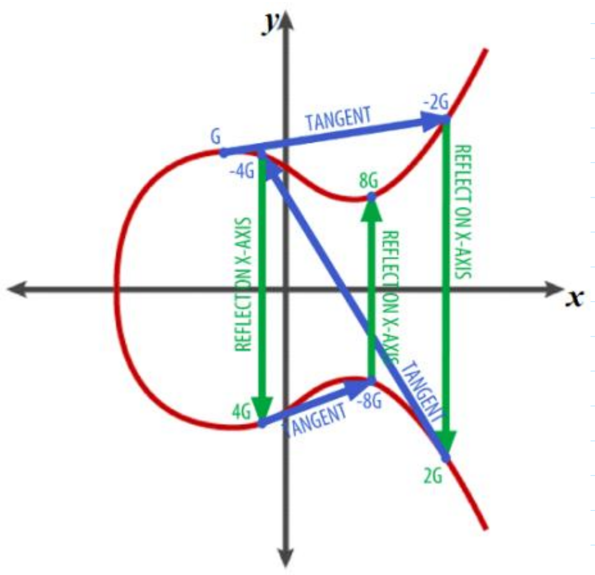
The positive and negative coordinates y and $-y$ in EC in the real numbers plane XOY are presented in Fig.
The positive and negative numbers for p=11 are presented in table .



y mod 11			$(-y)$ mod 11
1	odd	even	-1=10
2	even	odd	-2=9
3	odd	even	-3=8
4	even	odd	-4=7
5	odd	even	-5=6
6	even	odd	-6=5
7	odd	even	-7=4
8	even	odd	-8=3
9	odd	even	-9=2
10	even	odd	-10=1

Notice that performing operations **mod** p if y is odd then $-y$ is even and vice versa.
This property allows us to reduce bit representation of $\mathbf{PuK_{ECC}}=A=z*G=(x_A, y_A)$;
In normal representation of $\mathbf{PuK_{ECC}}$ it is needed to store 2 coordinates (x_A, y_A) every of them having 256 bits.
For $\mathbf{PuK_{ECC}}$ it is required to assign 512 bits in total.

Instead of that we can store only x_A coordinate with an additional information either coordinate y_A is odd or even.
The even coordinate y_A is encided by prefix 02 and odd coordinate y_A is encoded by prefix 03.
It is a compressed form of **PuK$_{ECC}$**.
If **PuK$_{ECC}$** is presented in uncompressed form than it is encoded by prefix 04.
Imagine, for example, that having generator G we are computing **PuK$_{ECC}$**=A=z∗G =(x_A, y_A) when z=8.
Please ignore that after this explanation since it is crasy to use such a small z. It is a gift for adversary
To provide a search procedure.
Then **PuK$_{ECC}$** is represented by point 8G as depicted in Fig. So we obtain a concrete point in EC being either even or odd.
The coordinate y_A of this point can be computed by having only coordinate x_A using formulas presented above and having prefix either 02 or 03.



EC: y^2=x^3+ax+b **mod** p
Let we computed **PuK$_{ECC}$**=A=(x_A, y_A)=8G .
Then $(y_A)^2 = (x_A)^3 + a(x_A) + b$ **mod** p is computed.
By extracting square root from $(y_A)^2$ we obtain 2 points:
8G and -8G with coordinates (**x**$_A$, **y**$_A$) and(**x**$_A$, -**y**$_A$).
According to the property of arithmetics of integers **mod** p
either **y**$_A$ is **even** and -**y**$_A$ is **odd** or **y**$_A$ is **odd** and -**y**$_A$ is **even**.
The reason is that **y**$_A$+(-**y**$_A$)=0 **mod** p as in the example above when p=11 and that there is a symmetry of EC with respect to x axis..
Then we can compress **PuK$_{ECC}$** representation with 2 coordinates (x_A, y_A) by representing it with 1coordinate x_A and adding prefix either 02 if y_A is even or 03 if y_A is odd.