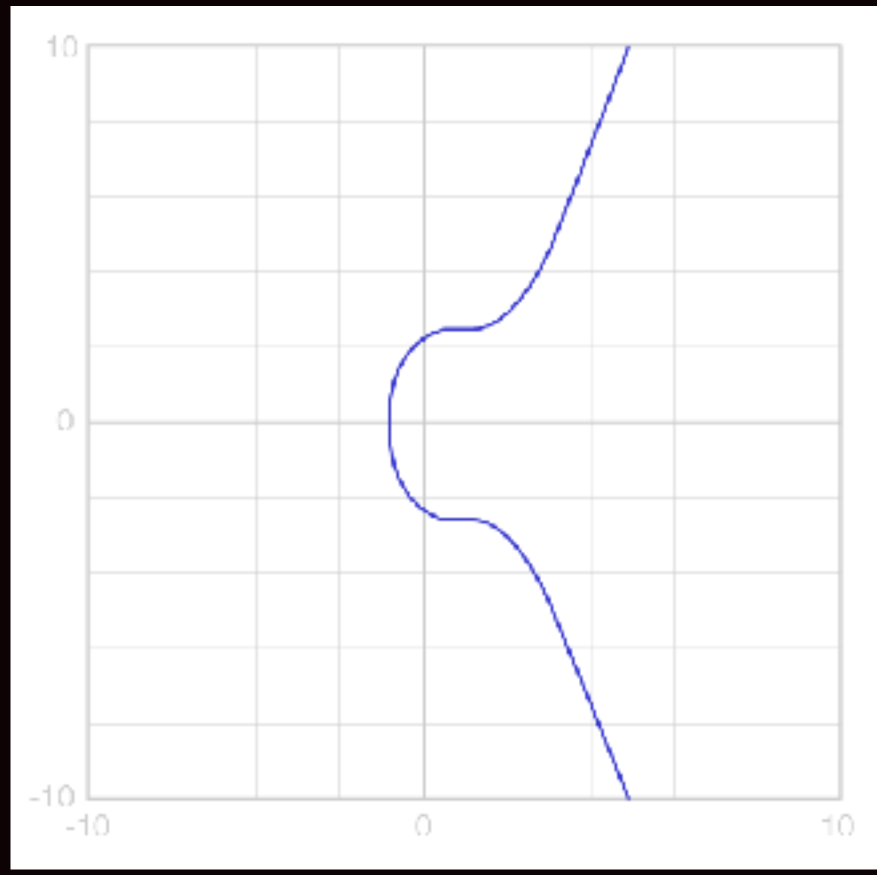
mobilefish.com

BLOCKCHAIN TUTORIAL 11

Elliptic Curve Key Pair Generation



$$y^2 = x^3 + ax + b$$

BLOCKCHAIN TUTORIAL 11

Elliptic Curve Key Pair Generation

ELLIPTIC CURVE KEY PAIR GENERATION

- Blockchain implementations such as Bitcoin or Ethereum uses Elliptic Curves (EC) to generate private and public key pairs.
- Elliptic Curve Cryptography (ECC) was invented by Neal Koblitz and Victor Miller in 1985.
- A 256-bit ECC public key provides comparable security to a 3072-bit RSA public key. The primary advantage of using Elliptic Curve based cryptography is reduced key size and hence speed.
- Elliptic curves have nothing to do with ellipses. Ellipses are formed by quadratic curves (x^2). Elliptic curves are always cubic (x^3).

STANDARDS FOR EFFICIENT CRYPTOGRAPHY GROUP

- The Standards for Efficient Cryptography Group (SECG) is an international consortium to develop commercial standards for efficient and interoperable cryptography based on elliptic curve cryptography (ECC).

- The SECG website is: http://www.secg.org

- The SECG has published a document with a recommended set of elliptic curve domain parameters, referred by the letters p, a, b, G, n, h. This data set { p, a, b, G, n, h } is collectively referred to as the Elliptic Curve Domain Parameters.

- These parameters have been given nicknames to enable them to be easily identified. For example: secp256k1

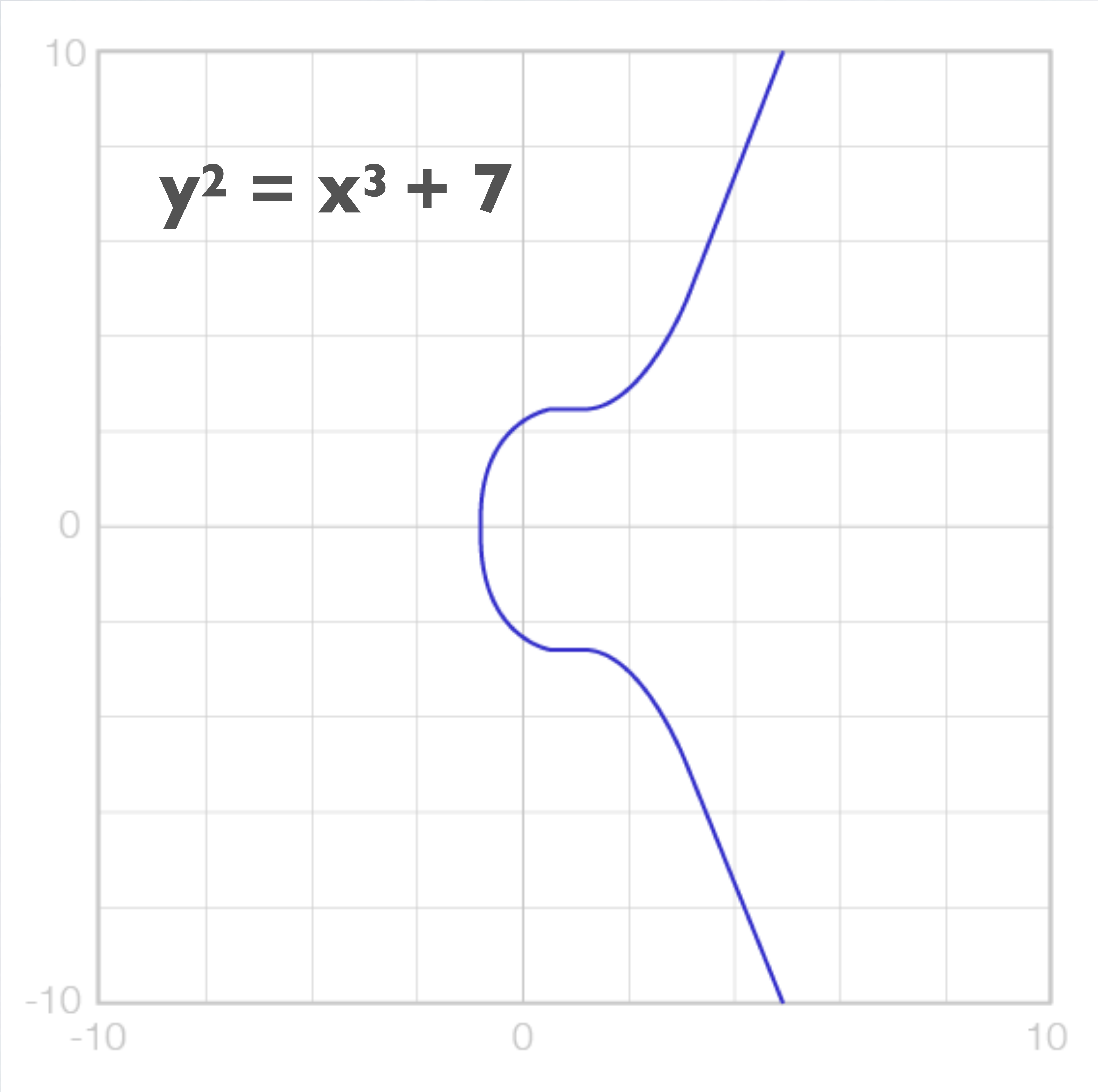
ELLIPTIC CURVE DOMAIN PARAMETERS

Parameters	Elliptic Curve Key Length	RSA Key Length
secp192k1	192	1536
secp192r1	192	1536
secp224k1	224	2048
secp224r1	224	2048
secp256k1	256	3072
secp256r1	256	3072
secp384r1	384	7680
secp512r1	512	15360

In this table you will find a set of elliptic curve domain parameters.

The elliptic curves uses smaller key sizes with respect to RSA providing comparable security.

SECP256K1



- Bitcoin and Ethereum both uses the same secp256k1 elliptic curve domain parameters.
- secp256k1 uses the following elliptic curve equation: $\mathbf{y^2 = x^3 + ax + b}$
- In the following slides we will go thru each parameter p, a, b, G, n, h
- Parameter a = 0
- Parameter b = 7

SECP256K1: PARAMETER P

- A finite field is a field with a finite number of elements, defined by parameter p, which is a prime number. Thus the finite field Fp = {0, ..., p - 1}

- This means that modulo p should be used in the equation:

  - The EC equation: $y^2 = x^3 + ax + b$

  - The EC equation with modulo operation: $y^2 = x^3 + ax + b \pmod p$

- $p = 2^{256} - 2^{32} - 2^9 - 2^8 - 2^7 - 2^6 - 2^4 - 1 = 2^{256} - 2^{32} - 977$

- p = FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFE FFFFFC2F

SECP256K1: PARAMETER G

- The basepoint G, also known as the generator or primitive element, is a predetermined point (X_G, Y_G) on the elliptic curve that everyone uses to compute other points on the curve.

- Often the basepoint G is displayed in two ways:

- Compressed form (prefix **02**):

  - **02** 79BE667E F9DCBBAC 55A06295 CE870B07 029BFCDB 2DCE28D9 59F2815B 16F81798

  - If the prefix is removed, the value is the X_G coordinate (= 79BE667E …)

  - To get the Y_G coordinate, calculate: $Y_G = (X^3_G + 7)^{½}$

SECP256K1: PARAMETER G

- Uncompressed form (prefix **04**):
  - **04 79BE667E** F9DCBBAC 55A06295 CE870B07 029BFCDB 2DCE28D9 59F2815B 16F81798 **483ADA77** 26A3C465 5DA4FBFC 0E1108A8 FD17B448 A6855419 9C47D08F FB10D4B8
  - If the prefix is removed, the first half of the value is the X_G coordinate (= **79BE667E** F9DCBBAC 55A06295 CE870B07 029BFCDB 2DCE28D9 59F2815B 16F81798)
  - The last half is the Y_G coordinate (= **483ADA77** 26A3C465 5DA4FBFC 0E1108A8 FD17B448 A6855419 9C47D08F FB10D4B8)

SECP256K1: PARAMETER N

- In my discrete logarithm video (part 9) I have explained what a cyclic group is. When you apply a certain number of operations to base point G, the cycle starts all over again in the same order. When the next cycle starts the first time it is indicated by parameter n which is called the order of base point G.

- n = FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFE BAAEDCE6 AF48A03B BFD25E8C D0364141

- The parameter n determines what the maximum value is that can be turned into a private key. Any 256-bit number in the range [1, n - 1] is a valid private key.

SECP256K1: PARAMETER H

- The parameter h is called the cofactor and has the constant value 1.
- Because it has value 1 it does not play a role in the key generation and I therefore will not elaborate on the purpose of this parameter.

DOT OPERATIONS

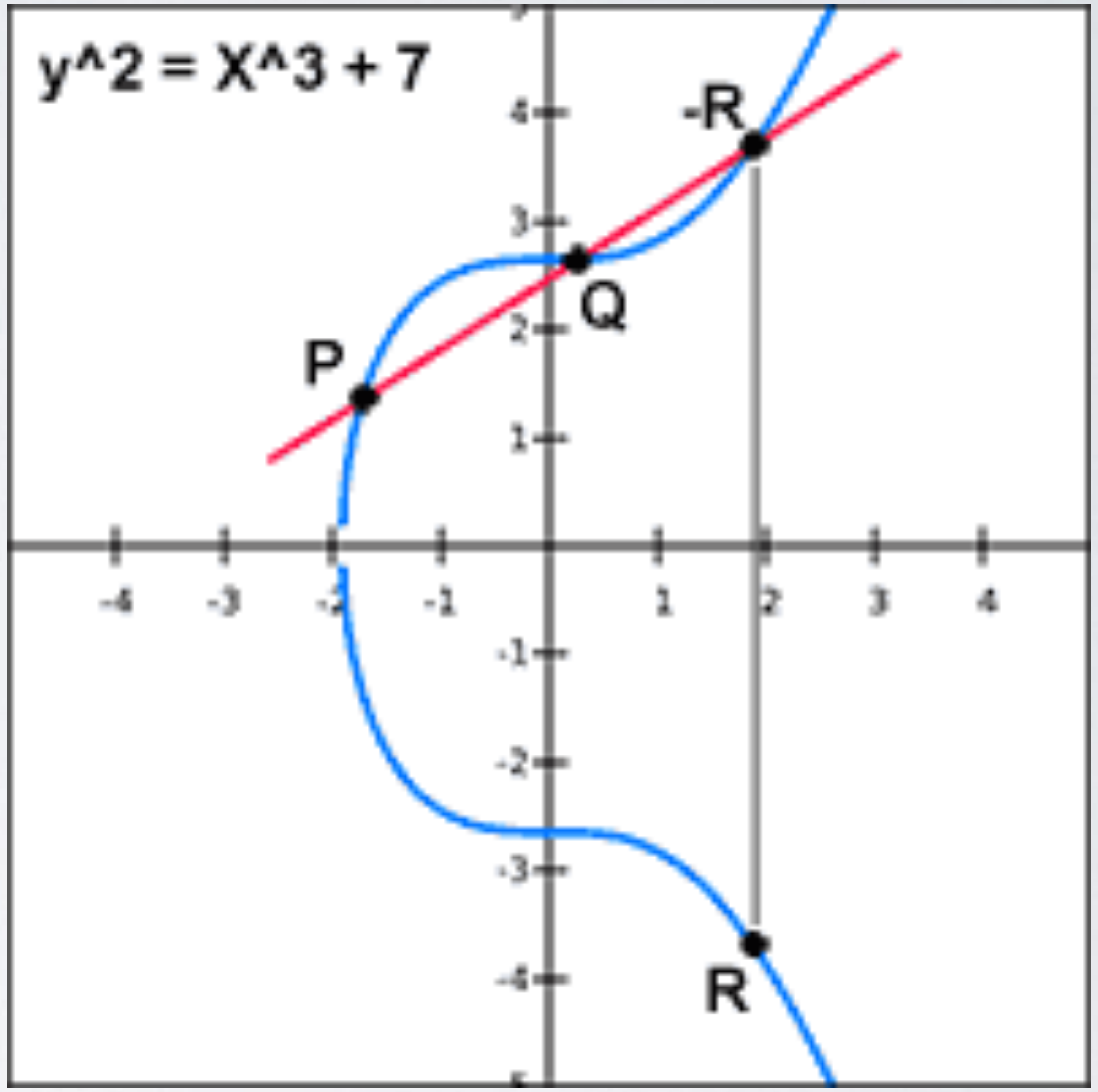
- There are two operations often called dot operations which can be applied to a base point (aka generator G) (x_G, y_G) on the elliptic curve:
    - Point addition
    - Point doubling
- The elliptic curve ($y^2 = x^3 + 7$) has the following properties:
    - If a line intersects two points P and Q, it intersects a third point on the curve -R.
    - If a line is tangent to the curve, it intersects another point on the curve.
    - All vertical lines intersects the curve at infinity.

POINT ADDITION

- Adding two points P and Q on a elliptic curve ($P \neq Q$).
- <u>Geometry approach:</u>
  - Draw a straight line between P (x_1, y_1) and Q (x_2, y_2).
  - The line will intersect the elliptic curve at exactly one more point -R.
  - The reflection of the point -R with respect to x-axis gives the point R (x_3, y_3), which is the results of addition of points P and Q.
- *Point addition* does not mean addition of the x or y coordinates of P and Q. It is just a name given for this approach.

POINT ADDITION

- Mathematical approach (ECAdd):
  - **$\lambda = (y_G - y)$ modinv$(x_G - x)$ (mod p)**
  - **$x_R = \lambda^2 - x - x_G$ (mod p)**
  - **$y_R = \lambda(x - x_R) - y$ (mod p)**
- λ is the slope of the line
- x and y are the coordinates of P
- Point Q is the base point G (x_G, y_G)



POINT DOUBLING

- Point doubling of point P on an elliptic curve. It is the same as moving point Q to same location as point P (P = Q)

- <u>Geometry approach:</u>

  - Draw a tangent line to the elliptic curve at point P.

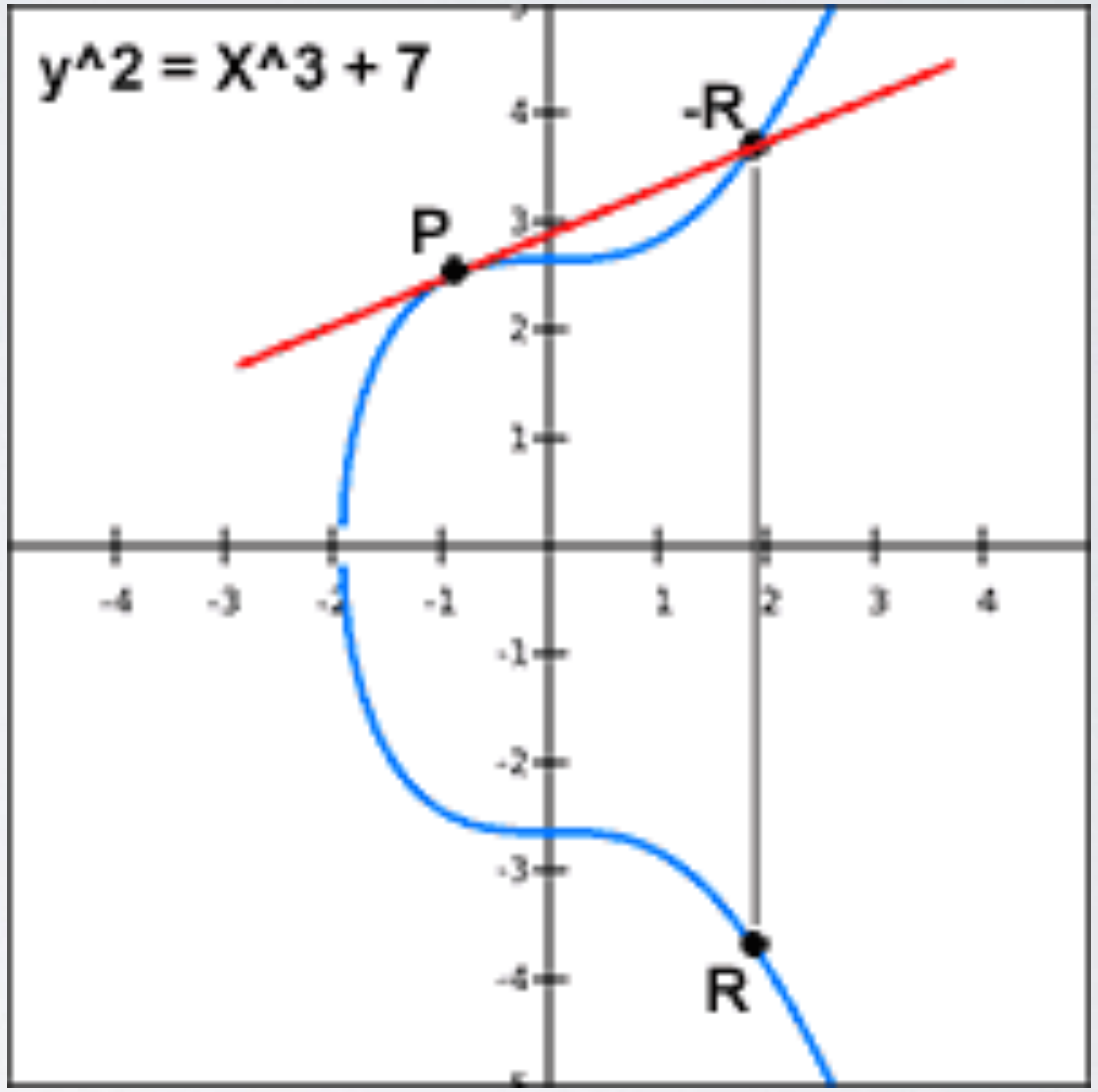
  - The line intersects the elliptic curve at the point -R.

  - The reflection of the point -R with respect to x-axis gives the point R, which is the results of doubling of point P.

- *Point doubling* does not mean multiplying the x or y coordinates of P. It is just a name given for this approach.

POINT DOUBLING

- Mathematical approach (ECDouble):
  - $\lambda = (3x^2)$ **modinv**$(2y)$ **(mod p)**
  - $x_R = \lambda^2 - 2x$ **(mod p)**
  - $y_R = \lambda(x - x_R) - y$ **(mod p)**
- λ is the slope of the line
- x and y are the coordinates of P



MATHEMATICAL EQUATIONS

ECAdd(x, y) {

$\lambda = (y_G - y)\ \mathrm{modinv}(x_G - x)\ (\mathrm{mod}\ p)$

$x_R = \lambda^2 - x - x_G\ (\mathrm{mod}\ p)$

$y_R = \lambda(x - x_R) - y\ (\mathrm{mod}\ p)$

}

ECDouble(x, y) {

$\lambda = (3x^2)\ \mathrm{modinv}(2y)\ (\mathrm{mod}\ p)$

$x_R = \lambda^2 - 2x\ (\mathrm{mod}\ p)$

$y_R = \lambda(x - x_R) - y\ (\mathrm{mod}\ p)$

}

More information about the mathematical equations can be found at:
http://www.mobilefish.com/services/cryptocurrency/cryptocurrency_v1.html

ADDITIONAL INFORMATION

- The following procedure describes how to generate a Bitcoin public key.
  For other blockchain implementations it may differ.

- When the "raw" Bitcoin public key is generated using the ECAdd and ECDouble functions it looks like this (large hexadecimal number):
  2A574EA59CAE80B09D6BA415746E9B031ABFBE83F149B43B37BE035B871648720
  336C5EB647E891C98261C57C13098FA6AE6821363C68FF15841B86DAD60241

- The actual Bitcoin address looks like: 1ADS8Lk6vN87Ri9hFjoFduPLNo76cwqUmf

- Additional conversion steps need to be applied on the "raw" Bitcoin public key to get the actual Bitcoin address which will be explained in another video.