ECDSA standards → Standards for Efficient Cryptography Group (SEC)
http://www.secg.org/
Bitcoin follows the **secp256k1** standard.
secp256k1: $y^2=x^3+a \cdot x+b \mod p$ $\quad p \sim 2^{256} \Rightarrow |p| = 256$ bits

Public Parameters: **PP**=(EC=**secp256k1**; BasePoint=***G***; finite fielf F_p of characteristic ***p***, ***p*** is prime;

$F_p = \{0, 1, 2, 3, \ldots, p-1\}$; $a, b \in F_p$

1. choose $x_1 \in F_p$
2. compute $x_1^3 + a \cdot x_1 + b = y_1^2$
3. compute $y_1 = \sqrt[2]{x_1^3 + a \cdot x_1 + b} = \sqrt[2]{y_1^2} = y_1$
4. The first coordinate of EC is (x_1, y_1)
5. choose $x_2 \in F_p$
6. ....

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.

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*** $\in F_p$.
F_p={0, 1, 2, 3, …, ***p***-1}; $+_{\mathbf{mod}\ p}$, $-_{\mathbf{mod}\ p}$, $\bullet_{\mathbf{mod}\ p}$, $:_{\mathbf{mod}\ p}$.

Example F_7 $\quad F_7 = \{0, 1, 2, 3, 4, 5, 6\}$; $p = 7$. Real $p \approx 2^{256}$.

Multiplication mod 7

*	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6
2	0	2	4	6	1	3	5
3	0	3	6	2	5	1	4
4	0	4	1	5	2	6	3
5	0	5	3	1	6	4	2
6	0	6	5	4	3	2	1

$8 \bmod 7 = 1$

$8 \mid 7$
$-7 \quad 1$
1

Addition mod 7

+	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	1						0
2							
3	3	4	5	6	0	1	2
4							
5							
6							

$\frac{5}{3} \bmod 7 = 5 \cdot 3^{-1} \bmod 7$

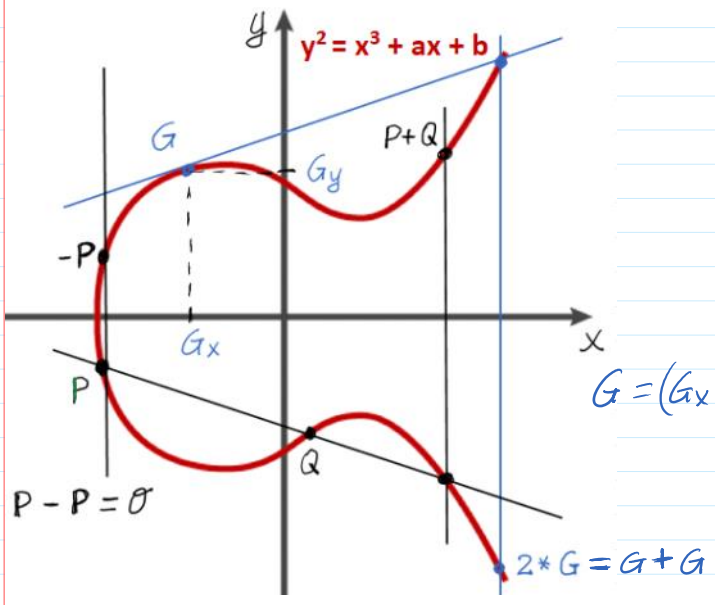
$3 \cdot 3^{-1} = \frac{3}{3} = 1 \bmod 7 \Rightarrow 3^{-1} = 5 \bmod 7$

$5 \cdot 3^{-1} = 5 \cdot 5 \bmod 7 = 4$

$2-3 \Rightarrow$ find -3
$2-3 = 2+(-3) \bmod 7$
$3+(-3) = 0 \Rightarrow -3 = 4$

$2-3 = 2+(-3) = 2+4 = 6 \bmod 7$

EC: $y^2 = x^3 + a \cdot x + b \bmod p7$



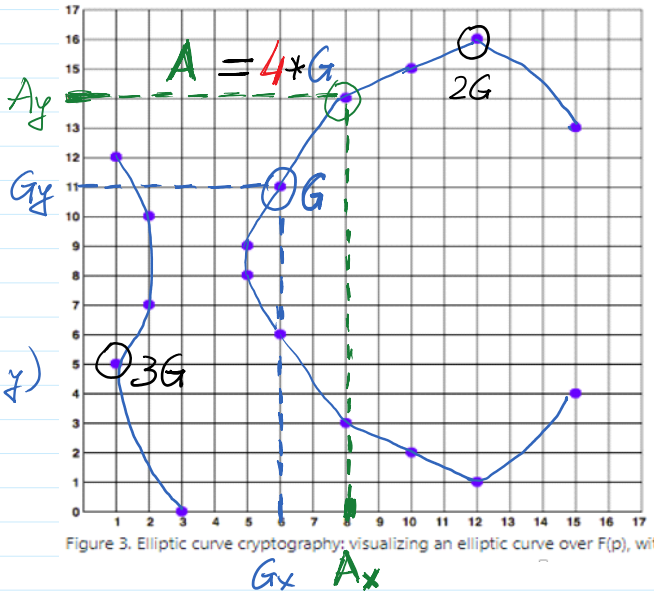


Figure 3. Elliptic curve cryptography: visualizing an elliptic curve over F(p), with p=17

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.

ECDSA animation

Signing and Verifying Ethereum Signatures – Yos Riady · Software Craftsman

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

Private Key of EC Cryptosystem (ECC) is **PrK $_{ECC}$=z**, where **z** is secret integer generated at random, i.e. **z** ← randi.
Public Key of ECC is **PuK$_{ECC}$=A=z*G**, e.g. let $z = 4 \Rightarrow A = (A_x, A_y)$
where * means **z**-times addition of points **G** in EC, i.e. multiplication of **G** by integer **z**.

Schnorr Signature

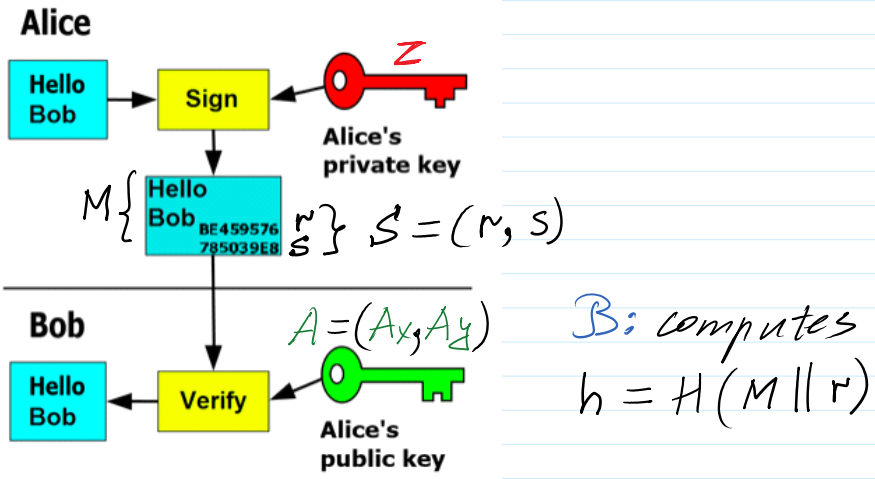
$PP = (p, g)$
$PrK = x \leftarrow randi\ (p-1)$
$PuK = a = g^x \bmod p$

ECDSA secp256k1

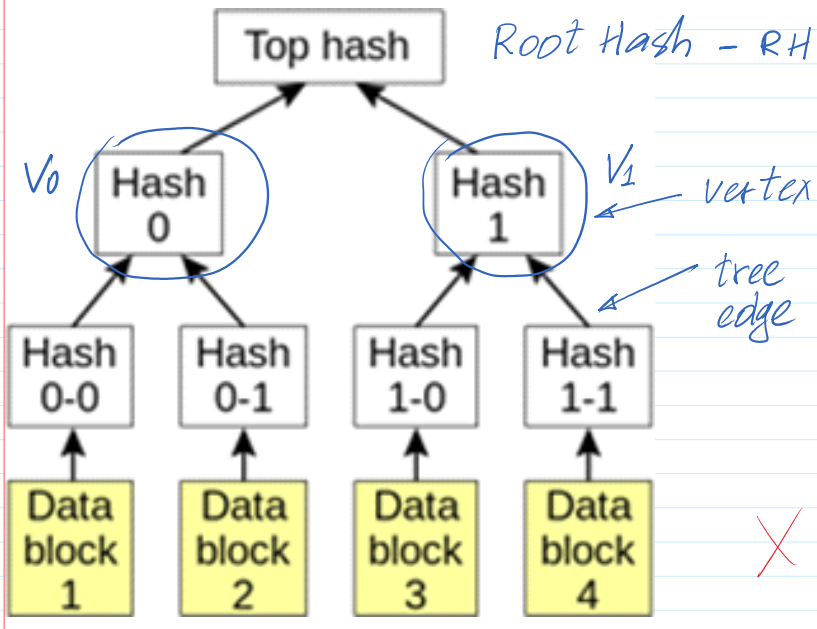
$PP = (secp256k1, G, p)$
$PrK = z \leftarrow randi\ (p-1)$
$PuK = A = z*G;\ A = (A_x, A_y)$

Asymmetric Signing - Verification
S=Sig(PrK$_A$, h)=(r, s)
V=Ver(PuK$_A$, S, h), V∈{True, False} ≡ {1, 0}

To sign message M A computes h-value of this message together with random parameter r: $h = H(M \| r)$



Bitcoin transactions are permanently recorded in the network through files called blocks.
Maximum size of the block is currently limited to 1 MB but it may be increased in the future.
Each block contains a UNIX time timestamp, which is used in block validity checks to make it more difficult for adversary to manipulate the block chain.
New blocks are added to the end of the record (block chain) by referencing the hash of the previous block and once added are never changed.
A variable number of transactions is included into a block through the merkle tree (fig 3.).
Transactions in the Merkle tree are hashed using double SHA256 (hash of the hash of the transaction message).



Transactions are included into the block's hash indirectly through the merkle root (top hash of a merkle tree).