Why Elliptic Curve Cryptography?

Table 6.1 Bit lengths of public-key algorithms for different security levels

Algorithm Family	Cryptosystems	Security Level (bit)			
		80	128	192	256
Integer factorization	RSA	1024 bit	3072 bit	7680 bit	15360 bit
Discrete logarithm	DH, DSA, Elgamal	1024 bit	3072 bit	7680 bit	15360 bit
Elliptic curves	ECDH, ECDSA	160 bit	256 bit	384 bit	512 bit
Symmetric-key	AES, 3DES	80 bit	128 bit	192 bit	256 bit

from Understanding Cryptography by Paar & Pelzl, pg 156

- Bitcoin uses ECDSA `https://en.bitcoin.it/wiki/Protocol_documentation#Signatures`
  Curve secp256k1 specified in `http://www.secg.org/sec2-v2.pdf`, pg 9
- amazon.com (currently) uses X25519, which is ECDHE with Curve25519
  Curve25519 uses a 256-bit key with 128-bits of security

1. Use Mathematica to draw the following elliptic curves

(a) Let $E : Y^2 = X^3 - 5X + 6$
Verify $4A^3 + 27B^2 \neq 0$
How many points on E are their own additive inverse?
i.e. How many points on E satisfy $P = -P$?

(b) $E : Y^2 = X^3 - 4X + 1$
Verify $4A^3 + 27B^2 \neq 0$
How many points on E are their own additive inverse?

(c) $E : Y^2 = X^3 - 3X + 2$
Verify $4A^3 + 27B^2 = 0$
By looking at the graph, why is this a problem for defining addition on E?

(d) $E : Y^2 = X^3$
Verify $4A^3 + 27B^2 = 0$
By looking at the graph, why is this a problem for defining addition on E?

2. Consider the elliptic curve $E : Y^2 = X^3 - 6X + 5$

(a) Verify that $P_1 = (-2, 3)$ and $P_2 = (2, 1)$ lie on E

(b) Use the geometric description of addition on E to find $P_1 + P_2$

(c) Use the geometric description of addition on E to find $2P_1$

(d) Use Theorem 6.6 to verify your answers to (b) and (c)

(e) Verify that $Q_1 = \left(\frac{1}{4}, -\frac{15}{8}\right)$ and $Q_2 = \left(\frac{58}{9}, \frac{413}{27}\right)$ lie on E

(f) Use Theorem 6.6 to find $Q_2 + Q_1$ and $Q_1 - Q_1$

Note: $-Q_1$ means the additive inverse of Q_1 in E