Elliptic Curve Cryptography

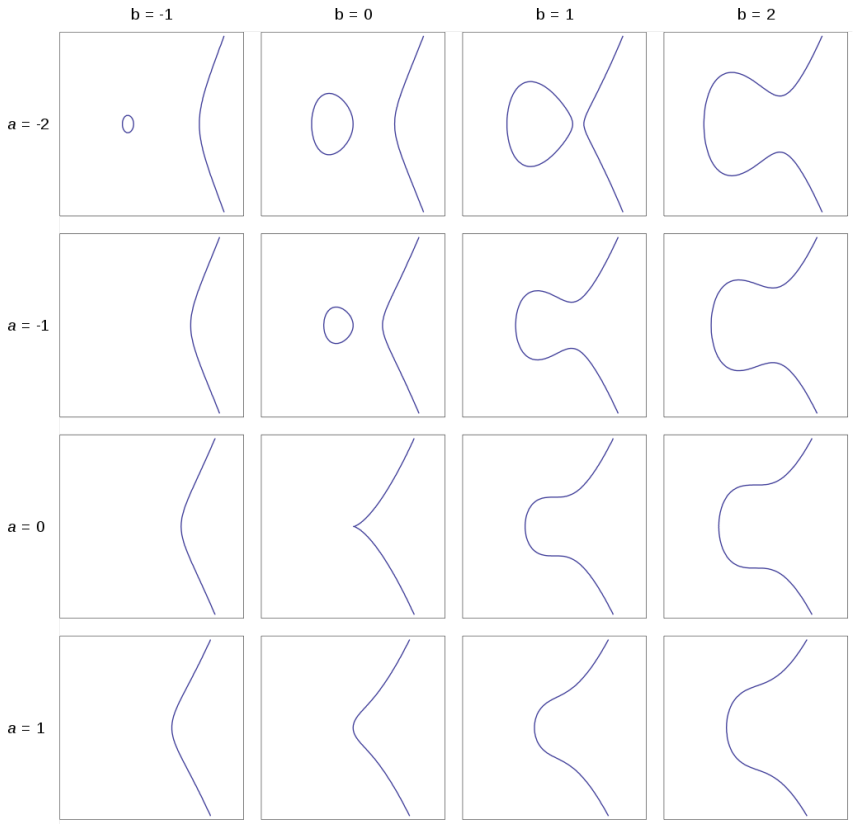
Meghana Doddapaneni

University of Maryland

5 December 2018

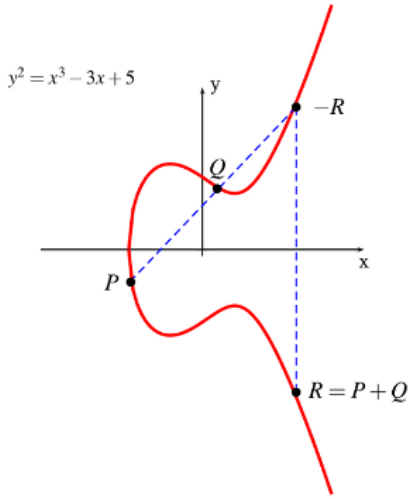
Introduction

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



en.wikipedia.org/wiki/Elliptic_curve

Elliptic Curve Addition



- $P = (x_p.y_p)$, $Q = (x_q, y_q)$, $R = (x_r.y_r) = P + Q$
- If $x_p \neq x_q$, then $x_r = s^2 - x_p - x_q$ and $y_r = y_p + s(x_r - x_p)$
- If $x_p = x_q$, then if $y_p = -y_q$, $P + Q = 0$ and if $y_p = y_q \neq 0$, then $s = \frac{3x_p^2 + a}{2y_p}$, $x_r = s^2 - 2x_p$, $y_r = y_p + s(x_r - x_p)$

www.embedded.com/design/safety-and-security/4396040/An-Introduction-to-Elliptic-Curve-Cryptography

Application to Cryptography

- Integers mod p $\longleftrightarrow$ Points on EC
- Multiplication mod p $\longleftrightarrow$ EC addition
- Exponentiation $\longleftrightarrow$ Integer times a point
- Discrete log problem (solve $g^k = h$ for k) $\longleftrightarrow$ EC discrete log problem (solve $kP = Q$ for k)

Curve

D.1.2.3 Curve P-256

$p =$ 115792089210356248762697446949407573530086143415290314195533631308867097853951

$n =$ 115792089210356248762697446949407573529996955224135760342422259061068512044369

$SEED =$ c49d3608 86e70493 6a6678e1 139d26b7 819f7e90

$c =$ 7efba166 2985be94 03cb055c 75d4f7e0 ce8d84a9 c5114abc af317768 0104fa0d

$b =$ 5ac635d8 aa3a93e7 b3ebbd55 769886bc 651d06b0 cc53b0f6 3bce3c3e 27d2604b

$G_x =$ 6b17d1f2 e12c4247 f8bce6e5 63a440f2 77037d81 2deb33a0 f4a13945 d898c296

$G_y =$ 4fe342e2 fe1a7f9b 8ee7eb4a 7c0f9e16 2bce3357 6b315ece cbb64068 37bf51f5

$$E : y^2 \equiv x^3 - 3x + b \mod p \text{ where}$$

$$b = 41058363725152142129326129780047268409114441015993725554835256314039467401291$$

$$p = 115792089210356248762697446949407573530086143415290314195533631308867097853951$$

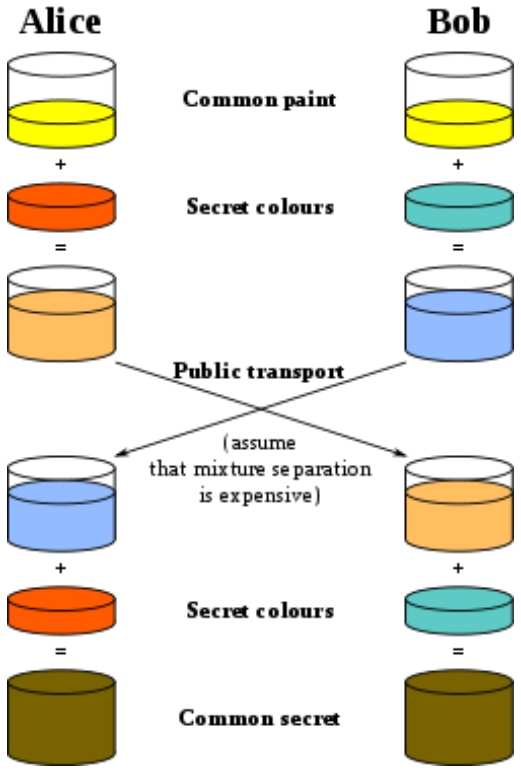
Example

Message: CATERPILLAR $\rightarrow$ 030120051816091212011800

Message point:

$$x = [030120051816091212011800,$$
$$25887690012340303386689306667201608449840332294974406626457147104170202682305]$$

Diffie Hellman Key Exchange



en.wikipedia.org/wiki/Diffie-Hellman_key_exchange

Diffie Hellman Key Exchange

- $N_a = 29$ (Alice's private key), $N_b = 19$ (Bob's private key)
- $N_a G = 29G$, $N_b G = 19G$
- $N_a N_b G = 19(29)G = N_b N_a G$

Diffie Hellman Key Exchange

Key found using Diffie Hellman:

$$\alpha = [771036971916402397351918762179597678664666730536106070673307489530058106459812399071861661971629677476277694870461969731130079454794767273776618931623370 6]$$

ElGamal Encryption

Non-EC version:

- Bob's public key: (modulus p, integer α, $\beta = \alpha^s \mod p$)
- $y_1 = \alpha^k \mod p$, $y_2 = x\beta^k \mod p$

Decryption: $x = y_1 y_2^{-s} \mod p$

ElGamal Encryption

- $s = 53$ (Bob's private key), $k = 67$ (Alice private key)
- $\beta = s\alpha$ (Bob publishes)
- $y_1 = k\alpha$, $y_2 = x + k\beta$

Decryption: $x = y_2 - sy_1$

ElGamal Encryption

Ciphertext found using EG:

$$y_1 = [92356132543430953744598233067113081659$$
$$45139445251043633490130785346016337097 0,$$
$$11190498677989925313066964928415002706 4$$
$$91770092962912428896827285737665564713 3]$$
$$y_2 = [10406900494706029739076126351427571155$$
$$33074553851412445941712277193962224226 65,$$
$$98872443134192988085933149412750334583 75$$
$$13184065198693222550525865030378227 55]$$

$$x = y_2 - sy_1$$

$$x = [03012005181609121201180 0,$$
$$25887690012340303386689306667201608449 8$$
$$40332294974406626457147104170202682305]$$

ElGamal Digital Signature

Non-EC version:
Signature:

- Private key (k, x), generator g
- $r = g^k \mod p$
- $s = (H(m) - xr)k^{-1} \mod p - 1$
- Signature (r, s)

Verification:

- $g^{H(m)} = g^{xr} r^s \mod p$

ElGamal Digital Signature

Signature:

- $k = 43$, $a = 23$ (Alice's private key)
- $R = kG = 43G$, $s = k^{-1}(m - ax) \mod N$
- Signed message: (m, R, s)

Verification:

- Alice's public key: (p, E, A, B)
- $V_1 = xB + sR$, $V_2 = mA$
- Verify that $V_1 = V_2$

ElGamal Digital Signature

Signature Verification:

$$
\begin{aligned}
V_1 = & [705021714611626904184653728457420172385958255328148780475348390204359849101083289011457934747844049138684738543981127868068211330857724980121626294957061 5] \\
& = V_2
\end{aligned}
$$

Comparison with RSA

- Key size versus cryptographic strength
  - A 2048 bit RSA key is equivalent to a 224 bit ECC key
  - A 3072 bit RSA key is equivalent to a 256 bit ECC key
- Key generation speed
  - 3072 bit RSA key $\rightarrow$ 9.8s
  - 283 bit ECC key $\rightarrow$ .27s
- RSA is easier to implement and much more widely used than ECC
- RSA has been much more studied than ECC