Elliptic Curve Cryptography

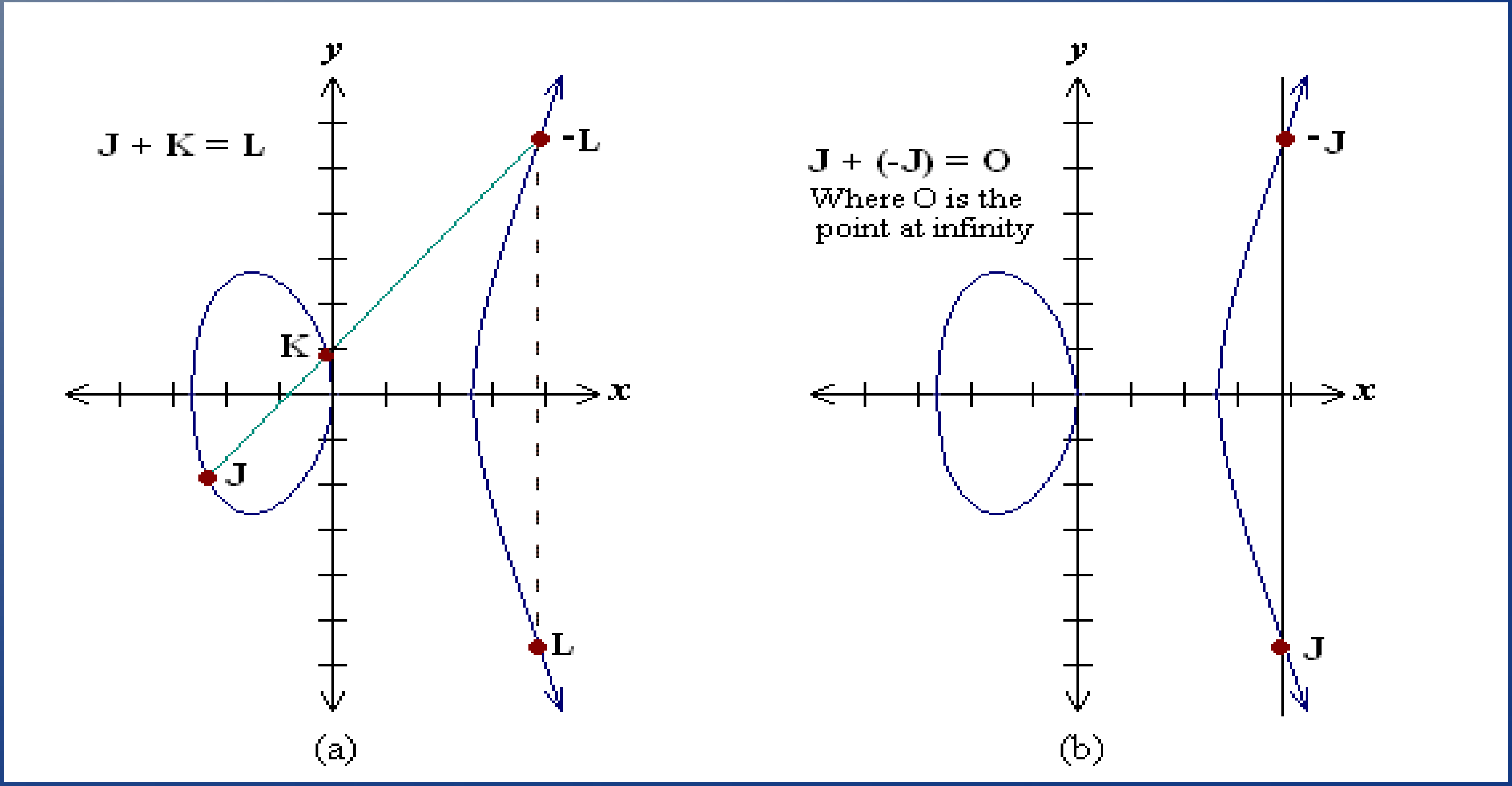
- Elliptic curve parameters over the finite field Fp
- T = (q, F R, a, b, G, n, h)
- q = the prime p
- a,b: the curve coeffiecient
- G: the base point (Gx,Gy)
- n: the order of G
- h: E(Fq)/n.
- Y^2 = x^3 + ax + b

Elliptic Curve Cryptography (ECC)

- ECC depends on the hardness of the discrete logarithm problem
- Let P and Q be two points on an elliptic curve such that kP = Q, where k is a scalar. Given P and Q, it is hard to compute k
- k is the discrete logarithm of Q to the base P.
- The main operation is point multiplication
- Multiplication of scalar k * p to achieve another point Q

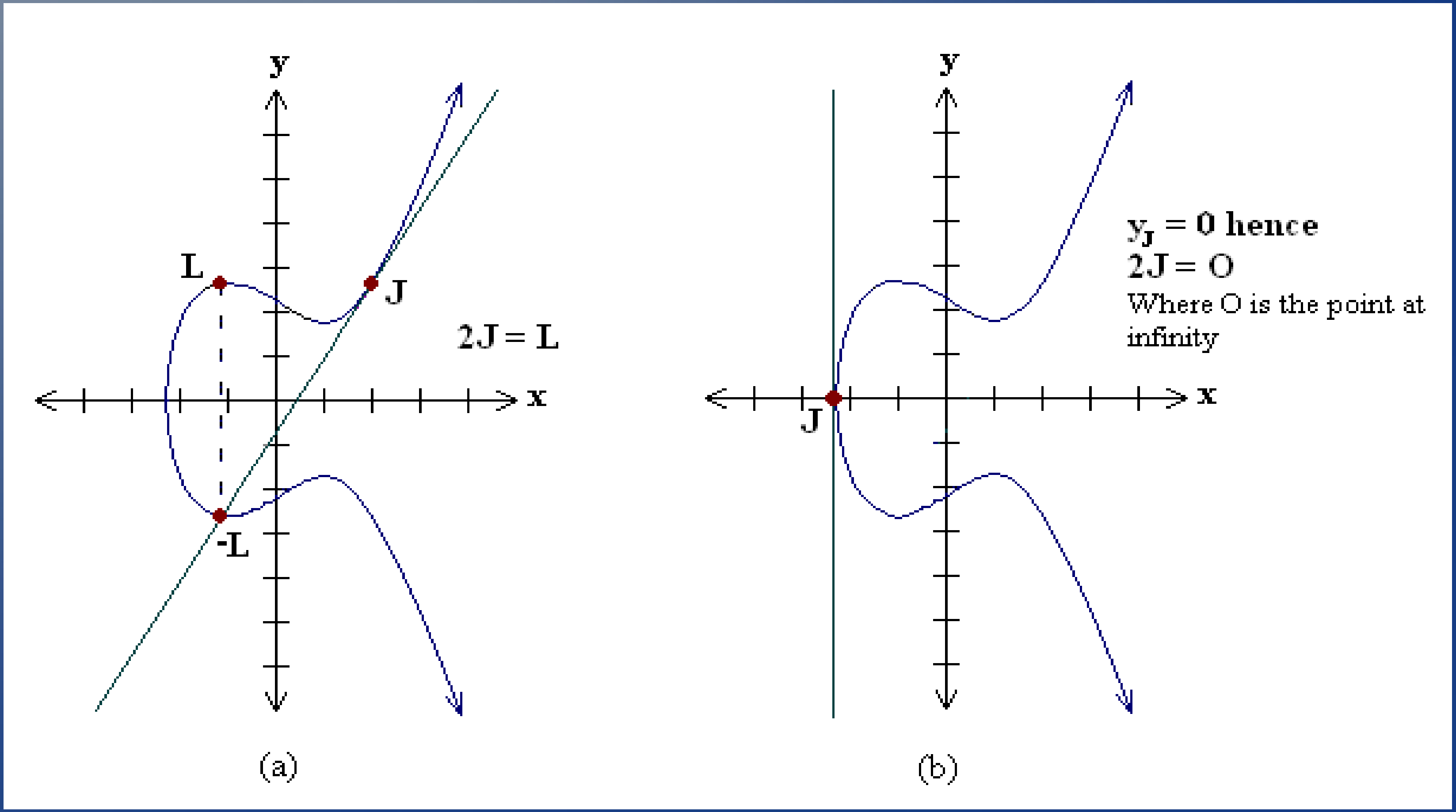
Point Addition

- Point addition is the addition of two points J and K on an elliptic curve to obtain another point L on the same elliptic curve.



Point Doubling

- Point doubling is the addition of a point J on the elliptic curve to itself to obtain another point L



Point Multiplication

- kP=Q
- Point multiplication is achieved by point addition and point doubling
- Point addition, adding two points J and K to obtain another point L i.e., L = J + K.
- Point doubling, adding a point J to itself to obtain another point L i.e. L = 2J.

Point Multiplication example

- Let k be a scalar that is multiplied with the point P to obtain another point Q on the curve. i.e. to find Q = kP.
- If k = 23 then kP = 23.P = 2(2(2(2P) + P) + P) + P
- As you can see point addition and point doubling are used to create Q
- The above method is called ‘double and add’ method for point multiplication
- Non-Adjacent Form and window Non-Adjacent Form are other methods

Elliptic Curve Digital Signature Algorithm Signing

- For signing a message m by sender A, using A's private key d

1. Calculate e = HASH (m), where HASH is a cryptographic hash function, such as SHA-1
2. Select a random integer k from [1,n − 1]
3. Calculate r = x1 (mod n),

   where (x1, y1) = k * G. If r = 0, go to step 2
4. Calculate s = k − 1(e + dr)(mod n). If s = 0, go to step 2
5. The signature is the pair (r, s)

Elliptic Curve Digital Signature Algorithm Verification

- For B to authenticate A's signature, B must have A’s public key Q

  1. Verify that r and s are integers in [1,n − 1]. If not, the signature is invalid

  2. Calculate e = HASH (m)

  3. Calculate w = s −1 (mod n)

  4. Calculate u1 = ew (mod n) & u2 = rw (mod n)

  5. Calculate (x1, y1) = u1*G + u2*Q

  6. The signature is valid if x1 = r(mod n)

Elliptic Curve Diffie Hellman

- a key pair consisting of a private key d (a randomly selected integer less than n, where n is the order of the curve, an elliptic curve domain parameter) and
- a public key Q = d * G (G is the generator point, an elliptic curve domain parameter).
- Let (dA, QA) be the private key - public key pair of A and (dB, QB) be the private key - public key pair of B
- its not possible to obtain the shared secret for a third party.

Elliptic Curve Diffie Hellman Pt. 2

1. The end A computes K = (xK, yK) = dA * QB
2. The end B computes L = (xL, yL) = dB * QA
3. Since dAQB = dAdBG = dBdAG = dBQA. Therefore K = L and hence xK = xL
4. Hence the shared secret is xK

- Since it is practically impossible to find the private key dA or dB from the public key K or L

Reason For Use

- Smaller key size
- Faster than RSA
- Good for handhelds and cell phones

Elliptic-Curve Digital Signature Algorithm (ECDSA)

NIST Guidelines for Public Key Sizes for AES			
ECC key size (bits)	RSA key size (bits)	Key size ratio	AES key size (bits)
163	1,024	1:6	
256	3,072	1:12	128
384	7,680	1:20	192
512	15,360	1:30	256

Supplied by NIST to ANSIX9F1

Table 1

NIST Reccomend Curves

- NIST reccomends p selections of 192,224,256,384,and 521 for use in government applications

Reference

[1] Darrel Hankerson, Julio Lopez Hernandez, Alfred Menezes, Software Implementation of Elliptic Curve Cryptography over Binary Fields, 2000, Available at http://citeseer.ist.psu.edu/hankerson00software.html

- [2] M. Brown, D. Hankerson, J. Lopez, A. Menezes, Software Implementation of the NIST
- Elliptic Curves Over Prime Fields, 2001, Available at
- http://citeseer.ist.psu.edu/brown01software.html
- [3] Certicom, Standards for Efficient Cryptography, SEC 1: Elliptic Curve Cryptography,
- Version 1.0, September 2000, Available at http://www.secg.org/download/aid-
- 385/sec1_final.pdf
- [4] Certicom, Standards for Efficient Cryptography, SEC 2: Recommended Elliptic Curve
- Domain Parameters, Version 1.0, September 2000, Available at
- http://www.secg.org/download/aid-386/sec2_final.pdf
- [5] Openssl, http://www.openssl.org
- [6] Certicom, http://www.certicom.com/index.php?action=ecc_tutorial,home
- [7] Alfred J. Menezes, Paul C. van Oorschot and Scott A. Vanstone, Handbook of Applied