Elliptic Curve Cryptography (ECC)

What is ECC

o What is Elliptic Curve Cryptography (ECC)?
- ECC:cryptography technique based on *elliptic curve theory* that can be used as faster, smaller, and more efficient cryptosystem.

o Who introduced it and when?
- Victor Miller and Neal Koblitz independently, around 1985

o What is the basic principle?
- Obtain same level of security as conventional cryptosystems but with much smaller key size

2 / 41

Why use ECC?

o How do we analyze Cryptosystems?
- How difficult is the underlying problem that it is based upon?
  - RSA – Integer Factorization
  - ElGamal - DSA – Discrete Logarithms
  - ECC - Elliptic Curve Discrete Logarithm problem
- How do we measure difficulty?
  - We examine the algorithms used to solve these problems

3 / 41

Benefits of ECC

o Same benefits of the other cryptosystems: confidentiality, integrity, authentication and non-repudiation but…

o ***Shorter key lengths***
- Encryption, Decryption and Signature Verification speed up
- Storage and bandwidth savings

4 / 41

Applications of ECC

o Many devices are small and have limited storage and computational power

o Where can we apply ECC?
- Wireless communication devices
- Smart cards
- Web servers that need to handle many encryption sessions
- Any application where security is needed but lacks the power, storage and computational power that is necessary for our current cryptosystems
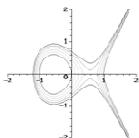
5 / 41

Equivalent key sizes

Symmetric	ECC	DH/DSA/RSA
80	163	1024
128	283	3072
192	409	7680
256	571	15360

6 / 41

Elliptic Curves

○ ○ ○

o An *Elliptic Curve* is such an alternate cyclic group. The group consists of all points of the form: $y^2 = x^3 + ax + b$. Where *x*, *y*, *a*, and *b* are all elements of a field *F*.
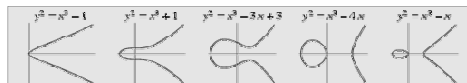
7 / 41

General form of a EC

○ ○ ○

o An *elliptic curve* is a plane curve defined by an equation of the form

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

Examples

8 / 41

Elliptic Curves over Finite Fields

○ ○ ○

o An elliptic curve **Group** over a finite field defines

● a **set of points** (*x, y*) that satisfy the elliptic curve equation, together with the "**point** *at infinity*" (**O**), the EC equation is given by:

• **GF(p):** $y^2 = x^3 + ax^2 + b$
  • *a, b* ∈ *GF(p)*, and
  • $4a^3 + 27b^2 \neq 0$ (mod p)

● A **group operation** → **Point addition**

9 / 41

Properties of EC Group Addition

○ ○ ○

1. **Commutative: P1 + P2 = P2 + P1**
2. **Identity (O):** **P + O = O + P = P**
3. ***Associative:***
   P1 + (P2 + P3) = (P1 + P2) + P3
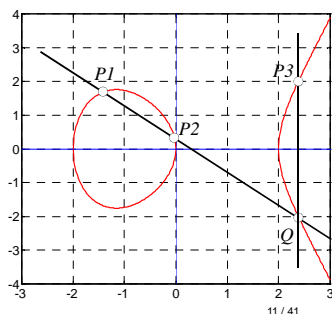4. ***Inverse:***
   there exists P' such that : P + P' = O

10 / 41

Point Addition

○ ○ ○

o **Adding 2 EC Points P1 & P2**:

● Draw straight line connecting P1 and P2

● Line intersects the EC at Q

● The *point* P3 = P1+P2 is the replica point of Q wrt x-axis.

● P1 + P2 = P3

11 / 41

Adding Example

○ ○ ○

o Suppose an elliptic curve *E* is defined as $y^2 = x^3 + 73$ and let *P1* =(2, 9) and P2 = (3, 10).
o The line **L** through *P1* and *P2* is $y = x + 7$
o Substituting into the elliptic equation for *E* yields $(x + 7)^2 = x^3 + 73$
o which yields
$x^3 - x^2 - 14x + 24 = 0$ ------ (1)
o We already know two roots, namely $x = 2$ and $x = 3$. Dividing (1) by $(x - 2)(x - 3)$ (= $x^2 - 5x + 6$) yields
$x + 4 => x = -4 => y = 3 => P3 = P1 + P2$
o *P3* = (-4, -3)

12 / 41

Point Doubling

o **What if $P_1 = P_2 = P$?**
- $P + P \rightarrow 2P$
- $\rightarrow$ point doubling
- Draw a tangent line through P,
- Tangent intersects the EC at R',
- The *point R* $=2P$ is the replica of R' wrt x-axis.
- $P+P = 2P = R$
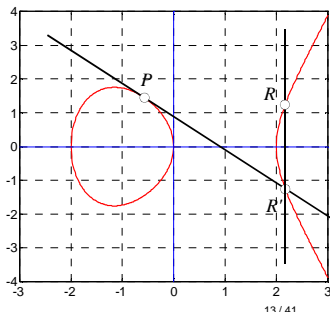
Doubling Example

To double the point $P = (-4, -3)$ (add it to itself).
o The slope of the tangent line to E at P can be obtained by differentiating the equation for E:
$2y\,dy = 3x^2\,dx \Rightarrow dy/dx = 3x^2/2y = -8$
o The tangent line is $y = -8(x + 4) - 3$
o Substituting the line equation into the equation for E
$(-8(x + 4) - 3)2 = x^3 + 73$ which yields
o The double root is $x = -4$
o It follows the third root $x = 72$
o $2R = 2P = (72, 611)$.

Scalar Multiplication

o Also called point multiplication
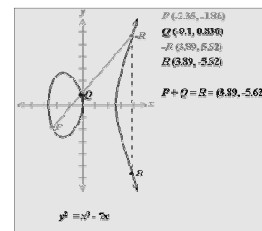- $KP = P + P + P + \ldots + P$ (K times)
- Where K is an integer.
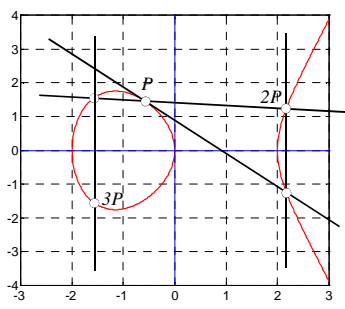
Sum of two points

Define for two points $P\,(x_1,y_1)$ and $Q\,(x_2,y_2)$ in the Elliptic curve

$$\lambda = \begin{cases} \dfrac{y_2 - y_1}{x_2 - x_1} & for \_ x_1 \neq x_2 \\ \dfrac{3x_1^2 + a}{2y_1} & for \_ x_1 = x_2 \end{cases}$$

Then $P+Q$ is given by $R(x_3,y_3)$:

$$x_3 = \lambda - x_1 - x_2$$
$$y_3 = \lambda(x_3 - x_1) + y_1$$

Point at infinity **O**

P+P = 2P

P and **-P** intersects with a third point at infinity point: **O (identity)**

As a result of the above case **P=O+P**

Hence all elliptic curves have additive identity **O**.

Example

E: $y^2 \equiv x^3 + 2x + 3 \bmod 5$
o The points on E are the pairs $(x, y) \bmod 5$ that satisfies the equation, along with the *point at infinity*.
o The possibilities for x are $\{0, 1, 2, 3, 4\}$
$x \equiv 0 \Rightarrow y2 \equiv 3 \bmod 5 \Rightarrow$ no solutions; *point O (0,0)*
$x \equiv 1 \Rightarrow y2 \equiv 6 \equiv 1 \bmod 5 \Rightarrow y \equiv 1, 4\ 5$
$x \equiv 2 \Rightarrow y2 \equiv 15 \equiv 0 \bmod 5 \Rightarrow y \equiv 0 \bmod 5$
$x \equiv 3 \Rightarrow y2 \equiv 36 \equiv 1 \bmod 5 \Rightarrow y \equiv 1, 4\ \bmod 5$
$x \equiv 4 \Rightarrow y2 \equiv 75 \equiv 0 \bmod 5 \Rightarrow y \equiv 0 \bmod 5$
o *E:* points are: (1,1), (1,4), (2,0), (3,1), (3,4),(4,0), *O*
$\rightarrow$ *#E (N) = 7*
o $(1,4)+(3,1) = (2,0)$ is also on the curve (closed).

Generator point

If the number of points (denoted as *r*) on the curve are equal to a prime integer, then we can find a *generator point* on the curve which generates *all the elliptic curve points.*

- It is possible to describe the discrete logarithm on a curve.
- If *r* is not a prime number, it is always possible to find a subgroup of elliptic curve points whose order is prime.
- In practice, *r* is chosen to be a multiple of a large prime (e.g. *r* = *kn* where *n* is a large prime and *k* is a *smooth* integer.)

19 / 41

Point Addition /Doubling Field Operations

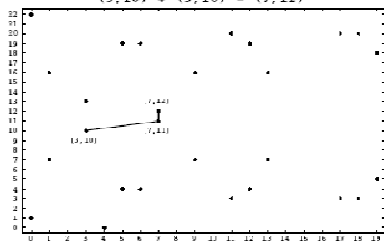- In *GF(p) ECC*

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

- Addition

$$x_3 = \lambda^2 - x_1 - x_2$$
$$y_3 = \lambda(x_1 - x_3) - y_1$$
$$where, \quad \lambda = (y_2 - y_1)/(x_2 - x_1)$$

- Doubling

$$x_3 = \lambda^2 - 2x_1$$
$$y_3 = \lambda(x_1 - x_3) - y_1$$
$$where, \quad \lambda = (3x_1^2 + a)/2y_1$$

(3,10) + (3,10) = (7,12)

21 / 41

## Finite Field Opetrations

- *GF(p)* (*Prime finite field*)
  - Elements are integers modulo *p* (0,1..*p*-1)

  - Operations are performed modulo *p*.

  - The prime number *p* is called the *modulus* of *GF(p)*.

Elliptic Curve Discrete Logarithm Problem (ECDLP)

Given an elliptic curve
and a basis point P, we can compute
   Q = KP
through k-1 iterative point additions.

Question: Is it possible to compute K when the point Q is known?

Answer: This is a hard problem known as the Elliptic Curve Discrete Logarithm (ECDLP).
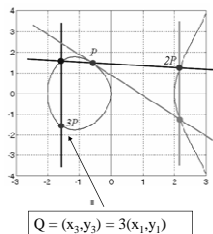
$Q = (x_3, y_3) = 3(x_1, y_1)$

23 / 41

The ECDLP is intractable; for a given field size, it is vastly harder to find K from KP and P than it is to find KP from K and P.

K is thus used as the private key; KP is used as the public.

The ECDLP is widely believed to be resistant to Number Field attacks. The best known attack is Pollard's Rho—whose difficulty grows more rapidly with the field size...

Scalar Multiplication Algorithms

○ *K* can be expanded to binary representation.

$K = k_{n-1}2^{n-1} + k_{n-2}2^{n-2} + \ldots + k_1 2 + k_0$

$KP = 2(2(\ldots2(2(k_{n-1}P)+k_{n-2}P)+\ldots)+k_1P)+k_0P$

| INPUT   K, P |
| OUTPUT   KP |
1. Initialize Q[0] = ∞ , Q[1] = P
2. **for** *i* = 0 **to** *n-1*
3.   **if** k[*i*] = 1 **then**
4.   Q[0] = ADD(Q[0],Q[1])
5.   **end if**
6.   Q[1] = DBL(Q[1])
7. **end for**
8. return Q[0]

| INPUT   K, P |
| OUTPUT   KP |
1. Initialize Q[0] = P
2. **for** *i* = n-2 **downto** 0
3.   Q[0] = DBL(Q[0])
4.   **if** k[*i*] = 1 **then**
5.   Q[0] = ADD(Q[0],P)
6.   **end if**
7. **end for**
8. return Q[0]

25 / 41

ECC Encryption/Decryption

■**Public Information**
  ■Elliptic Curve **E**, and the **base point** $P = (x_p, y_p)$.

Receiver
○ Choose a random Private key k_A and DECLARE $k_A P$ as a Public key

○ Compute $k_A C_1$ (= $k_A k_B P$)
○ Retrieve the message by computing:

$M = C_2 - k_A k_B P = C_2 + (- k_A k_B P)$

Sender
○ Message *M* is embedded into *E*.
○ Choose a random Private key k_B
○ Compute:
  • $k_B k_A P$.
 • $C_1 = k_B P$
  • $C_2 = (x_m, y_m) + k_B k_A P$
○ Send (C_1 , C_2) as the encrypted message.

26 / 41

Projective Coordinate Systems

○ In *GF(p) ECC* One inversion operation costs
  • 9 to 30 multiplications
    for 100 bits or more field elements

Transferring the point coordinates into another coordinates that can eliminate the intermediate inversions is very important requirement

Affine point → Projective point → Scalar Multiplication → Projective point → Affine point
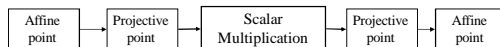
27 / 41

ECDH
Elliptic Curve Diffie-Hellman

ECDH is analogous to conventional Diffie-Hellman; p(qG) = q(pG); qG, pG are public values; p and q are private.

ECDSA: Elliptic Curve Digital Signature Algorithm

ECDSA is analogous to DSA:
  Key generation
    signature
      verification