# COM5336 Cryptography
# Lecture 11
# Elliptic Curve Cryptography

Scott CH Huang

## Outline

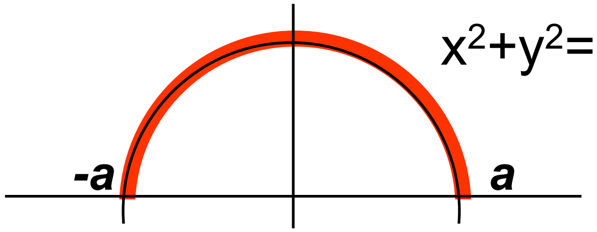
- Elliptic Curve (橢圓函數)
  - Basic definition
  - Operations：addition & scalar
- Elliptic Curve Cryptosystem
  - Security considerations
    - ECC vs. RSA comparison
  - Implementation  considerations
  - Applications
- Conclusions

## Holy Roman Empire vs Elliptic Curve Cryptosystem

- "*The Holy Roman Empire is neither holy, nor Roman, nor an empire*."~ Voltaire
- The Elliptic Curve Cryptosystem is neither related to ellipses nor itself a cryptosystem.

# The Arc Length of an Ellipse

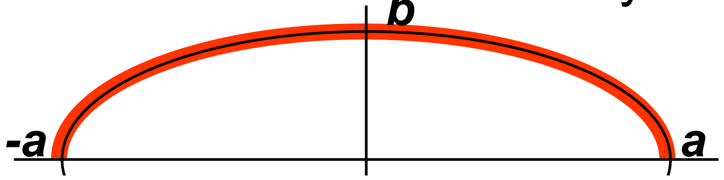
The arc length of a half **circle** is given by the familiar integral



$x^2+y^2=a^2$

$$\boxed{\int_{-a}^{a} \frac{a\,dx}{\sqrt{a^2-x^2}}}$$

The arc length of a half **ellipse** is more complicated



$x^2/a^2 + y^2/b^2 = 1$

$$\boxed{\int_{-a}^{a} \sqrt{\frac{a^2-\left(1-b^2/a^2\right)x^2}{a^2-x^2}}\,dx}$$

Let $k^2 = 1 - b^2/a^2$ and change variables $x \rightarrow ax$. Then the arc length of an ellipse is

$$a\int_{-1}^{1}\sqrt{\frac{1-k^2x^2}{1-x^2}}\,dx \qquad \boxed{a\int_{-1}^{1}\frac{1-k^2x^2}{\sqrt{(1-x^2)(1-k^2x^2)}}\,dx}$$

$$\text{Arc Length} = a\int_{-1}^{1}\frac{1-k^2x^2}{y}\,dx$$

**An Elliptic Curve!**

with $y^2 = (1 - x^2)(1 - k^2x^2)$ = quartic in $x$.

An **elliptic integral** is an integral $\int R(x,y)\,dx$ , where $R(x,y)$ is a rational function of the coordinates $(x,y)$ on an "elliptic curve"

$E : y^2 = f(x)$ = cubic or quartic in $x$.

# Elliptic Curve Cryptography

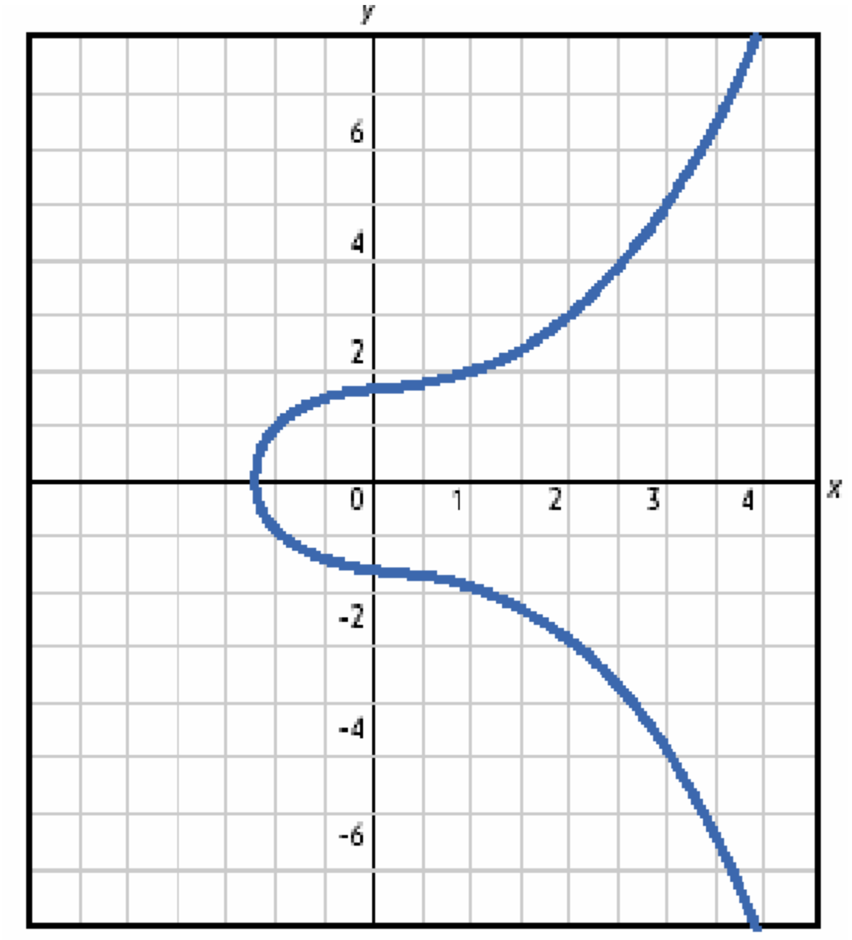
- ECC itself is NOT a cryptosystem
- It uses EC to define a **group**.
- We can use this group to form an Elliptic Curve Discrete Log Problem (ECDLP)
- Any DLP-based system can replace its $Z_p$ by the EC group

## ECC (cont)

- Thus, we can use EC on all DLP-based encryption/decryption/key exchange/signature algorithms: Diffie-Hellman, ElGamal, ElGamal signature, DSA
- There are no good attacks on ECDLP now, so ECC can use very short keys.
  - ECC uses 160-bit keys (cf RSA/ElGamal 1024 bits)

# Elliptic Curves

- Elliptic curves are not ellipses (the name comes from elliptic integrals)
- Circle
  - $x^2 + y^2 = r^2$
- Ellipsis
  - $a \cdot x^2 + b \cdot y^2 = c$
- Elliptic curve
  - $y^2 = x^3 + a \cdot x + b$

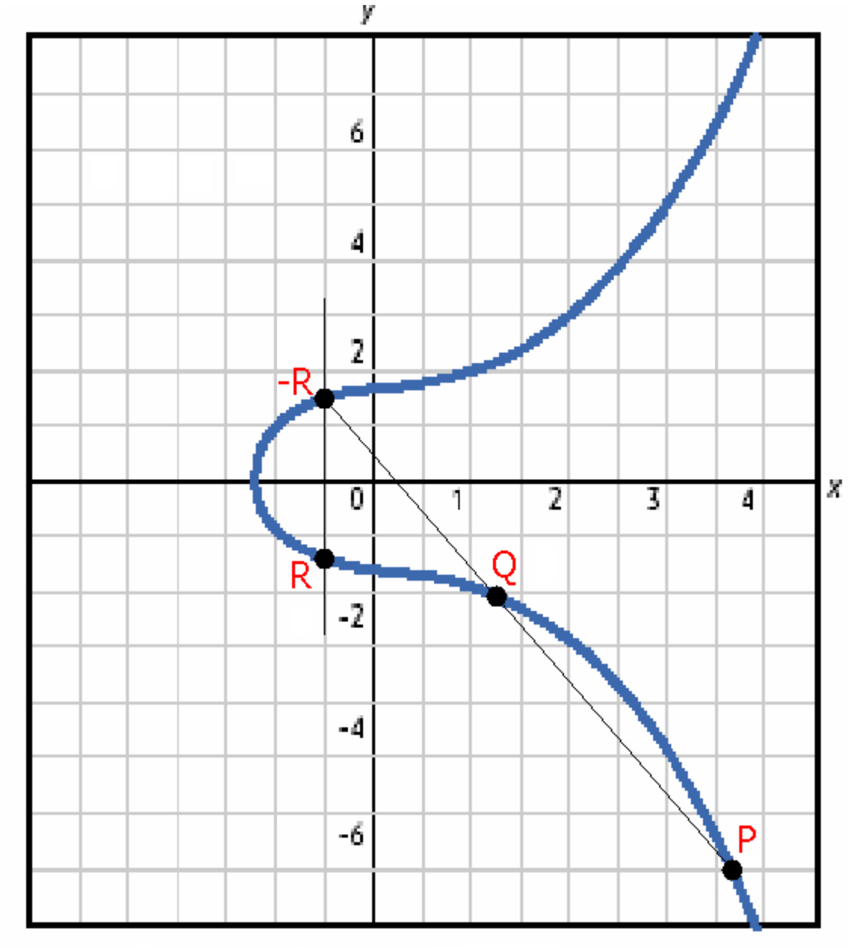


## Elliptic Curves Over Real Numbers

- An elliptic curve over reals is the set of points (x,y) which satisfy the equation $y^2 = x^3 + a \cdot x + b$, where x, y, a, and b are real numbers
- If $4 \cdot a^3 + 27 \cdot b^2$ is not 0 (i.e. $x^3 + a \cdot x + b$ contains no repeated factors), then the elliptic curve can be used to form a group
- An elliptic curve group consists of the points on the curve and a special point O (point at infinity)
- Elliptic curves are additive groups
  - Addition can be defined geometrically or algebraically

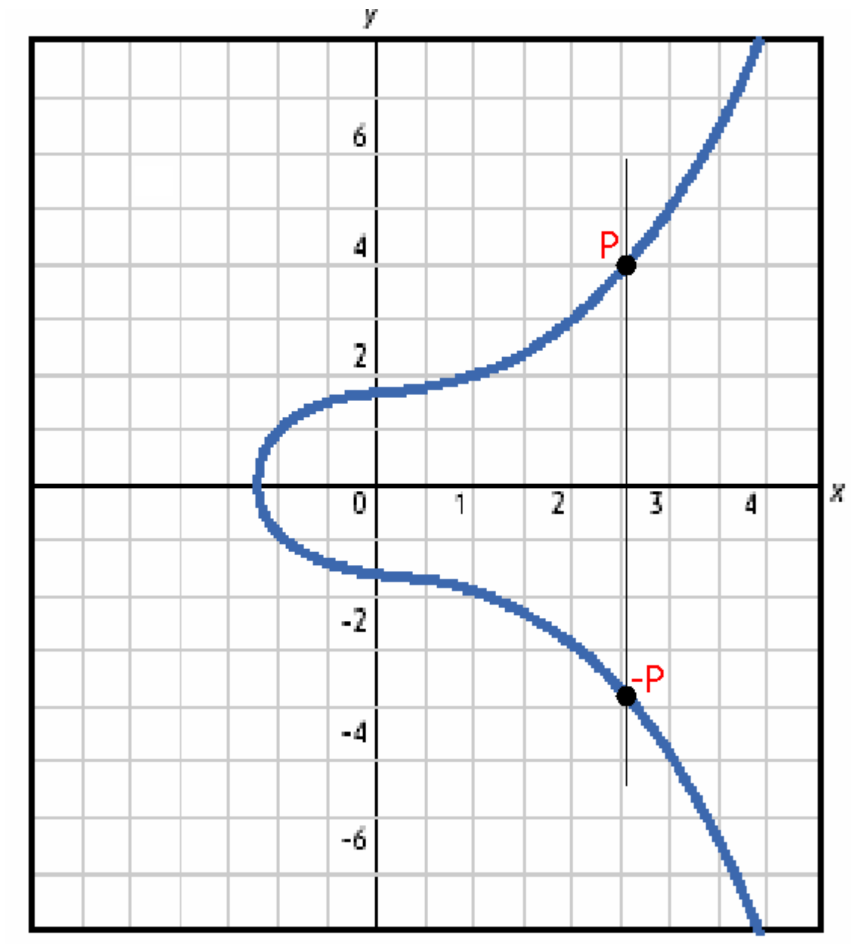
# Adding Points P and Q

- Draw a line that intersects distinct points P and Q
  - The line will intersect a third point -R
- Draw a vertical line through point -R
  - The line will intersect a fourth point R
- Point R is defined as the sum of points P and Q
  - R = P + Q



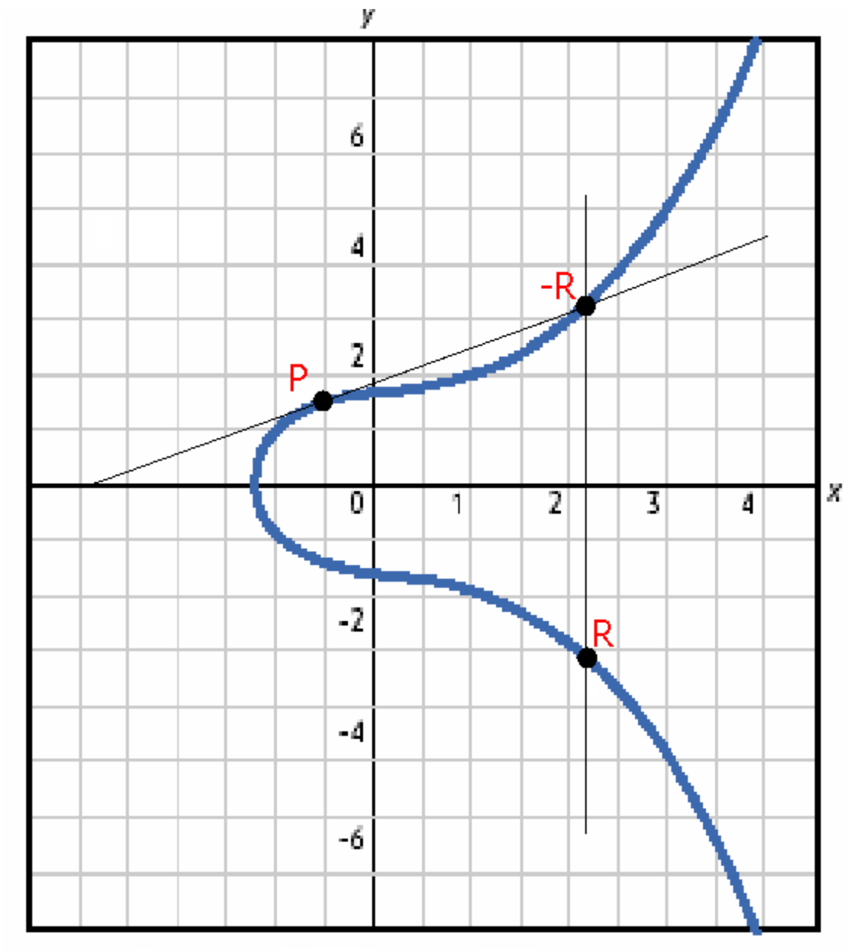
# Adding Points P and -P

- Draw a line that intersects points P and -P
  - The line will not intersect a third point
- For this reason, elliptic curves include O, a point at infinity
  - P + (-P) = O
  - O is the additive identity



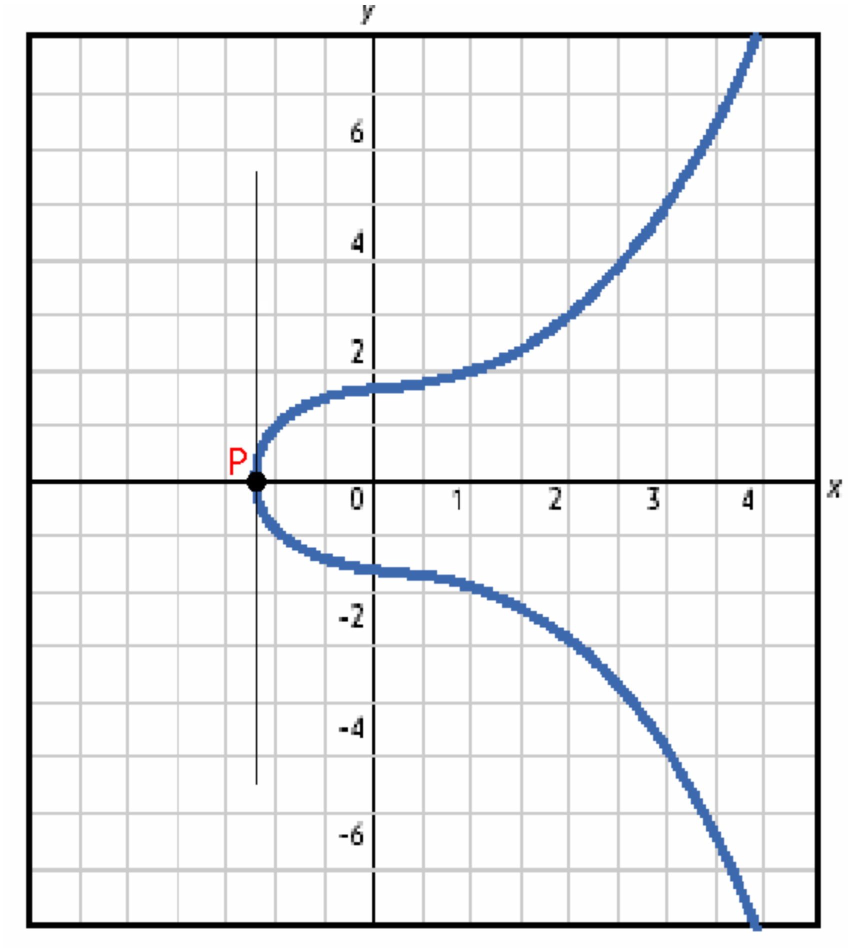
# Doubling the Point P

- Draw a line tangent to point P
  - The line will intersect a second point -R
- Draw a vertical line through point -R
  - The line will intersect a third point R
- Point R is defined as the summation of point P with itself
  - R = 2·P

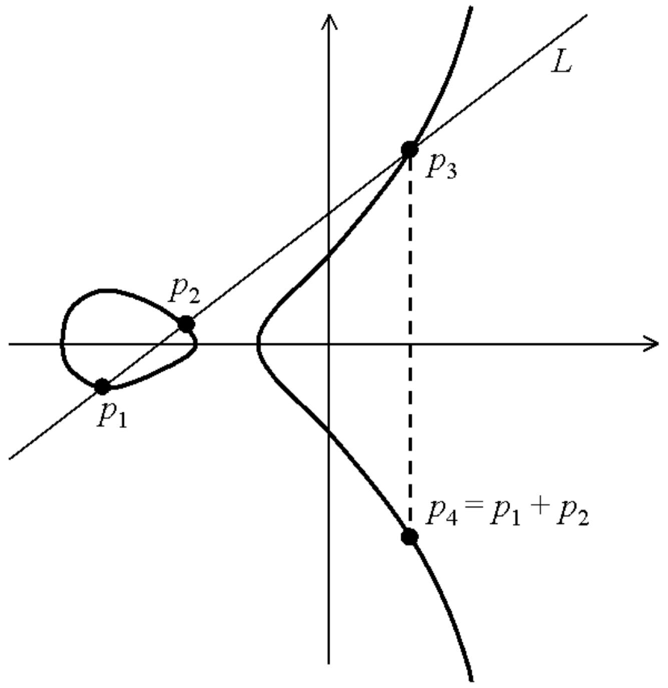


# Doubling the Point P if yP = 0

- Draw a line tangent to point P
  - If yP = 0, the line will not intersect a second point
- 2·P = O when yP = 0
  - 3·P = P *(2·P + P)*
  - 4·P = O *(2·P + 2·P)*
  - 5·P = P *(2·P + 2·P + P)*

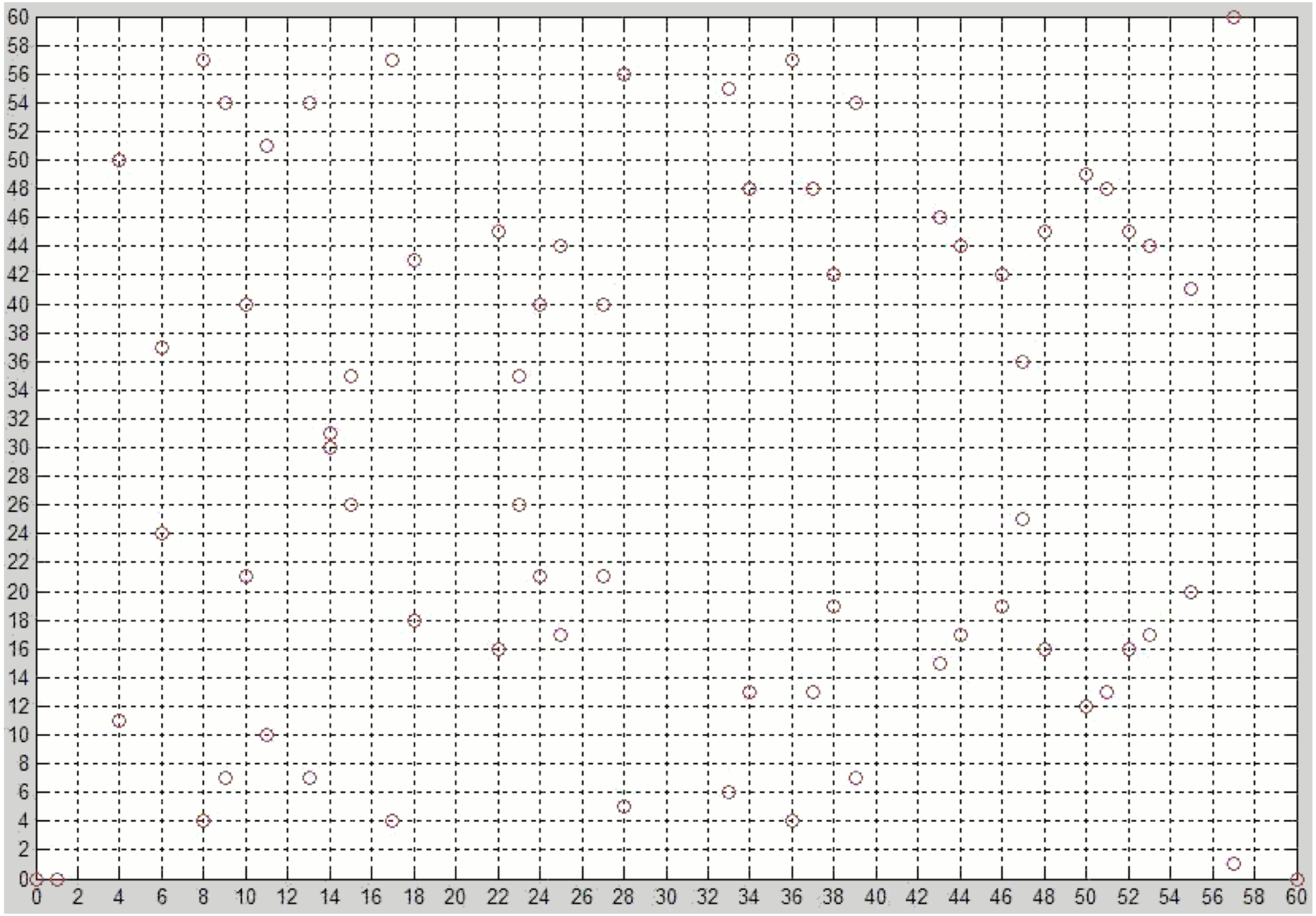


# Some Elliptic Curve over Reals



## Elliptic Curves over GF(61)

$$y^2 \equiv x^3 - x \pmod{61}$$



# Algebraic Approach

- Addition: $P=(x_1,y_1)$, $Q=(x_2,y_2)$, $R=(x_3,y_3) = P + Q$
  - $\lambda = (y_2 - y_1) / (x_2 - x_1)$
  - $x_3 = \lambda^2 - x_1 - x_2$
  - $y_3 = -y_1 + \lambda( x_1 - x_3)$
- Doubling: $R=(x_3,y_3) = 2P$
  - $\lambda = (3x_1^2 + a) / (2y_1)$
  - $x_3 = \lambda^2 - 2x_1$
  - $y_3 = -y_1 + \lambda( x_1 - x_3)$

## Characteristic of a Ring

- Let $\mathcal{R}$ be a ring. The characteristic of $\mathcal{R}$, denoted by char($\mathcal{R}$), is defined to be the smallest number *n* such that $\underbrace{a+\ldots+a}_{n \text{ times}} = 0$ for all $a \in \mathcal{R}$.

- If $\mathcal{R}$ contains 1, then the characteristic is the smallest number *n* such that $\underbrace{1+\ldots+1}_{n \text{ times}}=0$

- If such number do not exists, then char($\mathcal{R}$) is defined to be 0.

## Elliptic Curves Over Finite Fields

- Calculations with real numbers are slow and rounding causes inaccuracy
- Speed and accuracy are important for cryptography
- Use elliptic curve groups over the finite field $F_p$ or $F_2{}^m$

## Elliptic Curves Over Finite Fields

- Because it’s a finite field, a finite number of points make up the curve
  - This means there is **no curve anymore**
  - But also no more rounding
- Geometric definitions of addition and doubling don’t work on these curves
- Algebraic definitions still hold

## Elliptic Curves in Weierstrass Form

- $y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$
- All coefficients and variables are assumed to be in a field **F**.
- Depending on the characteristic of **F**, different forms of elliptic curves are used.
- In particular, if the char(**F**)= 2, special treatment is necessary.
- Fields with characteristic 3 are usually not important in applications and are often omitted.

## Elliptic Curves over F w/ char(F)$\neq$2,3

- Elliptic curve over F : $y^2 = x^3 + ax + b$,
  where $x, y, a, b \in \mathbf{F}$.
- $4a^3 + 27b^2 \neq 0$ (i.e. $x^3 + ax + b$ contains no repeated factors).
- Everything is the same as the elliptic curve over reals except that there's no curve anymore and their definitions are purely algebraic.
- As usual, the elements are the (x,y) pairs satisfying the above equation along with ***O***.

## Algebraic Definition of Addition, Inverse, and Doubling

- Inverse: $P=(x_1,y_1)$
  - $-P=(x_1,-y_1)$
- Addition: $P=(x_1,y_1)$, $Q=(x_2,y_2)$, $R=(x_3,y_3) = P + Q$
  - $\lambda = (y_2 - y_1) / (x_2 - x_1)$
  - $x_3 = \lambda^2 - x_1 - x_2$
  - $y_3 = -y_1 + \lambda( x_1 - x_3)$
- Doubling: $R=(x_3,y_3) = 2P$
  - $\lambda = (3x_1^2 + a) / (2y_1)$
  - $x_3 = \lambda^2 - 2x_1$
  - $y_3 = -y_1 + \lambda( x_1 - x_3)$

## Elliptic Curves Over F w/ char(F)=2

- Usually we only consider two types of elliptic curves: *zero j-invariant* & *nonzero j-invariant*.
- Zero j-invariant : $y^2 + a_3 y = x^3 + a_4 x + a_6$, where $x, y, a_3, a_4, a_6 \in F$.
- Nonzero j-invariant : $y^2 + xy = x^3 + a_2 x^2 + a_6$, where $x, y, a_3, a_4, a_6 \in \mathbf{F}$.

## Inverse, Addition, & Doubling of Zero j-invariant Elliptic Curves

**Zero j-invariant Elliptic Curve: $y^2 + a_3 y = x^3 + a_4 x + a_6$**

- Inverse: $P=(x_1,y_1)$
  - $-P=(x_1,-y_1+a_3)$
- Addition: $P=(x_1,y_1)$, $Q=(x_2,y_2)$, $R=(x_3,y_3) = P + Q$
  - $\lambda = (y_1 + y_2) / (x_1 + x_2)$
  - $x_3 = \lambda^2 + x_1 + x_2$
  - $y_3 = \lambda( x_1 + x_3) + y_1 + a_3$
- Doubling: $R=(x_3,y_3) = 2P$
  - $\lambda = (x_1^2 + a_4) / a_3$
  - $x_3 = \lambda^2$
  - $y_3 = \lambda( x_1 + x_3) + y_1 + a_3$

## Inverse, Addition, & Doubling of Nonzero j-invariant Elliptic Curves

**Nonzero j-invariant Elliptic Curve: $y^2 + xy = x^3 + a_2x^2 + a_6$**

- Inverse: $P=(x_1,y_1)$
  - $-P=(x_1,y_1+x_1)$
- Addition: $P=(x_1,y_1)$, $Q=(x_2,y_2)$, $R=(x_3,y_3) = P + Q$
  - $\lambda = (y_1 + y_2) / (x_1 + x_2)$
  - $x_3 = \lambda^2 + \lambda + x_1 + x_2 + a_2$
  - $y_3 = \lambda( x_1 + x_3) + x_3 + y_1$
- Doubling: $R=(x_3,y_3) = 2P$
  - $x_3 = (a_6 / x_1^2) + x_1^2$
  - $y_3 = x_1^2 + ( x_1 + y_1/x_1) x_3 + x_3$

# Number of Points on $\mathcal{E}$(F)

- **#$\mathcal{E}$(F):** number of points on **$\mathcal{E}$(F).**
- **#$\mathcal{E}$(F)** represents how many different pieces of information can be coded.
- **Hesse's Theorem**: $|\#\mathcal{E}(\mathbf{F}_p)-(p+1)|\leq 2\sqrt{p}$
- **The Weil Conjecture**:

  Let $t=p+1-\#\mathcal{E}(\mathbf{F}_p)$, then $\#\mathcal{E}(\mathbf{F}_{p^k})=p^k+1-\alpha^k-\beta^k$,

  where $1-tx+px^2=(1-\alpha x)(1-\beta x)$.
- If the Weil Conjecture is true, then we can determine $\#\mathcal{E}(\mathbf{F}_{p^k})$ based on $\#\mathcal{E}(\mathbf{F}_p)$.

## Operations

| $Z_p^*$ | $\mathcal{E}(F)$ |
|---|---|
| Multiplication ↔ | Addition |
| Exponentiation ↔ | Scalar (Multiple) |

Elliptic curves are interesting because they provide a way of constructing "elements" and "rules of combining" that produce groups.

These groups have enough familiar properties to build cryptographic algorithms, but they don't have certain properties that may facilitate cryptanalysis. For example, there is no good notion of "smooth." That is, there is no set of small elements in terms of which a random element has a good chance of being expressed by a simple algorithm. Hence, index calculus discrete logarithm algorithm do not work.

Elliptic curves over the finite field GF($2^n$) are particularly interesting. The arithmetic processors for the underlying field are easy to construct and are relatively simple to implement for n in the range of 130 to 200. These systems have the potential to provide small and low-cost public-key cryptosystems. Many public-key algorithms, like Diffie-Hellman, ElGamal, and Schnorr, can be implemented in elliptic curves over finite fields.

## Correspondence between $Z_p^*$ and $E(\mathbf{F})$ notation

| Group | $Z_p^*$ | $\mathcal{E}(\mathbf{F})$ |
|---|---|---|
| Group elements | Integers $\{1, 2, \ldots, p\text{-}1\}$ | Points $(x, y)$ on $\mathcal{E}(\mathbf{F})$plus $O$ |
| Group operation | Multiplication modulo $p$ | Addition of points |
| Notation | Elements：$g$, $h$<br>Multiplication：$g \bullet h$<br>Inverse：$g^{-1}$<br>Division： $g / h$<br>Exponentiation： $g^a$ | Elements：$P$, $Q$<br>Addition：$P+Q$<br>Negative：$-P$<br>Subtraction：$P-Q$<br>Multiple：$aP$ |
| Discrete Logarithm Problem | Given $g \in Z_p^*$ and $h = g^a \bmod p$, find $a$ | Given $P \in \mathcal{E}(\mathbf{F})$ and $Q = aP$, find $a$ |

## Elliptic Curve Cryptosystem

- Proposed by Neal Koblitz and V. S.  Miller (independently)in 1985.They did not invent a cryptographic algorithm using ECs.
- Standards：
  - ANSI X9.62 — ECDSA
  - ANSI X9.63 — EC Key Agreement and Transport Protocols
  - IEEE P1363 standard—Standard Specifications for PKI
- Hard problem：
  - Elliptic curve discrete logarithm problem (ECDLP)
    - Given $P$, $Q$ to find k such that $Q$=k$P$

## ECC Security Considerations

- Attacks：
  - Attacks on the elliptic curve discrete logarithm problem
    - Naive exhaustive search
    - Pohlig-hellman algorithm
    - Baby-step Giant-step algorithm：$(n)^{1/2}$ steps
    - Pollard's rho algorithm：$(\pi n/2)^{1/2}$ steps
    - Parallelized Pollard's rho algorithm：$(\pi n)^{1/2}/2r$ steps
  - Attacks on the hash function employed
  - Other attacks
- Experimental Results

The fastest so far

Parallelized Pollard's rho algorithm：$(\pi n)^{1/2}/2r$ steps
(r processors, n is the order of the point P)

## Comparison：ECC vs. RSA

| ECC | | RSA | |
|---|---|---|---|
| Key length (bits) | Breaking time (MIPS-Years) | Key length (bits) | Breaking time (MIPS-Years) |
| 150 | $3.8 \times 10^{10}$ | 512 | $3 \times 10^{4}$ |
| 205 | $7.1 \times 10^{18}$ | 768 | $2 \times 10^{8}$ |
| 234 160 | $1.6 \times 10^{28}$ | 1024 | $3 \times 10^{11}$ |
| | | 1280 | $1 \times 10^{14}$ |
| | | 1536 | $3 \times 10^{16}$ |
| | | 2048 | $3 \times 10^{20}$ |
| ***Pollard rho method to attack ECDLP*** | | ***Generalized number field sieve method to attack factorization*** | |

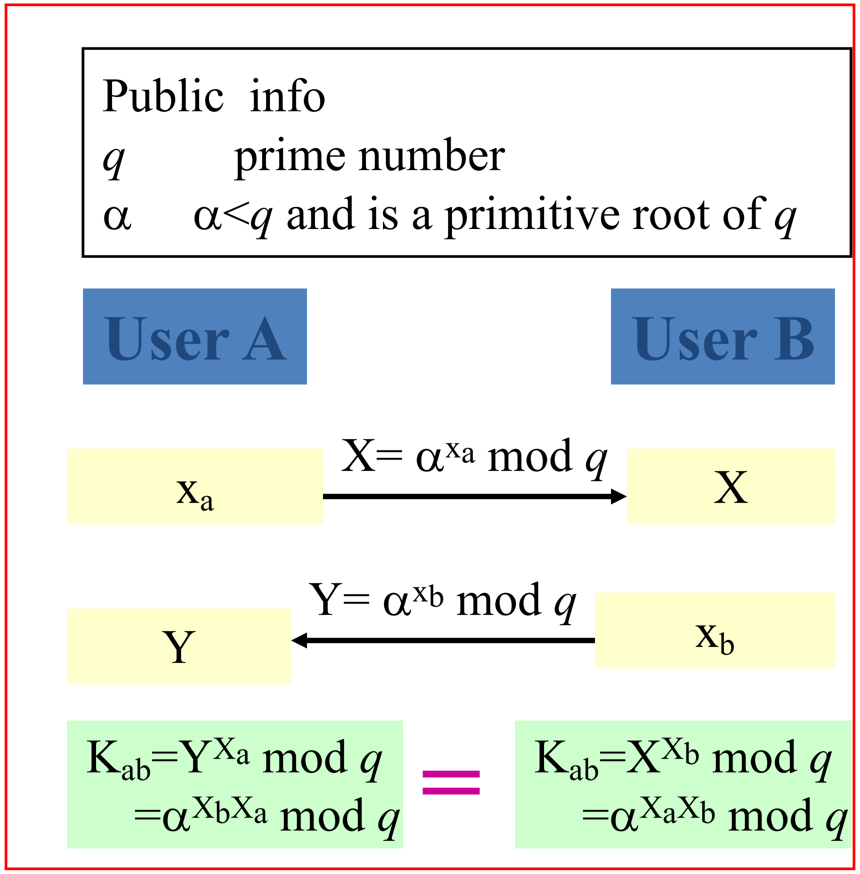
## ECC Implementation Considerations

- Security considerations
- Suitability of methods available for optimizing finite field arithmetic(addition, multiplication, squaring, and inversion)
- Suitability of methods available for optimizing elliptic curve arithmetic (point addition, point doubling, and scalar multiplication)
- Application platform (software, hardware, or firmware)
- Constraints of a particular computing environment (e.g., processor speed, storage, code size, gate count, power consumption)
- Constraints of a particular communications environment (e.g., bandwidth, response time)

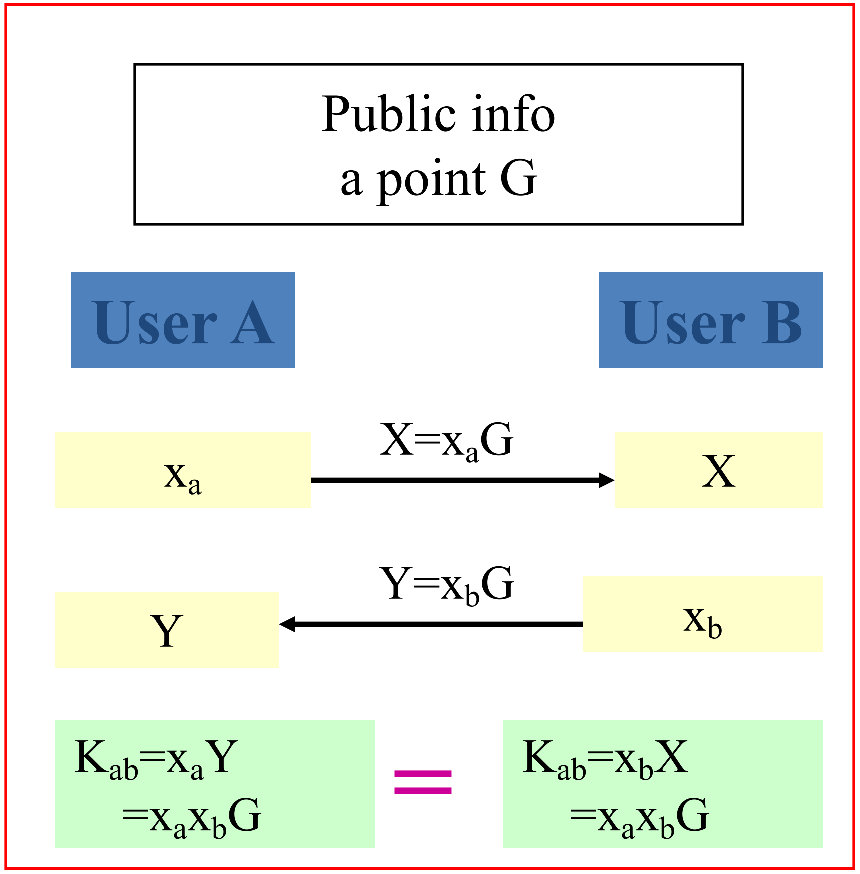
## Applications

- Key Exchange (Diffie Hellman)
- Encryption & Decryption (ElGamal)
- Digital Signature (ElGamal, DSA)
- Authentication

# EC Diffie-Hellman Key Exchange


Public info
$q$ prime number
$\alpha$ $\alpha<q$ and is a primitive root of $q$
User A
User B
$x_a$
$X = \alpha^{x_a} \bmod q$
$X$
$Y$
$Y = \alpha^{x_b} \bmod q$
$x_b$
$K_{ab} = Y^{X_a} \bmod q = \alpha^{X_b X_a} \bmod q$
=
$K_{ab} = X^{X_b} \bmod q = \alpha^{X_a X_b} \bmod q$

*Diffie-Hellman Key Exchange Algorithm*



*Key Exchange Using Elliptic Curve Cryptography*

# EC ElGamal Encryption

| | ElGamal encryption | EC ElGamal encryption |
|---|---|---|
| Key generation | Generate a prime $p$ of length k bits.<br>Compute a generator $g$ for $Z_p^*$<br>Select $x$, $2 \le x \le p-2$<br>$y = g^x \bmod p$<br>public key：$(g, p, y)$<br>private key：$x$ | Select $(p, a, b, G)$, $G \in \mathcal{E}(\mathbf{F})$ such that the smallest $n$ for $nG=O$ is large.<br>Select $n_a<n$<br>$P_a=n_aG$<br>public key：$(p, a, b, P_a)$<br>private key：$n_a$ |
| Encryption | Select $k$, $1 \le k \le p-2$<br>$C=\{g^k \bmod p, My^k \bmod p\}$ | Select $n_k< n$.<br>$C_m = \{n_kG, P_m+n_kP_a\}$ |
| Decryption | Receive $C = \{a, b\}$<br>$M = b/a^x \bmod p$ | Receive $C_m = \{P_k, P_b\}$<br>$P_m=P_b- n_aP_k = P_m+n_kP_a-n_aP_k$ |

## DSA vs. ECDSA

| | DSA | ECDSA |
|---|---|---|
| Key generation | Select $p$, $q$, $x$ and $q \mid p$-1, $1 \le d < q$.<br>Select $h \in Z_p^*$ and compute $g = h^{(p-1)/q} \bmod p$ until $g \ne 1$.<br>$y = g^d \bmod p$<br>public key：($p$, $q$, $g$, $y$)<br>private key：$d$ | Select $E$ over $\mathcal{E}(\mathbf{F})$, select $d$, $1 \le d < n$.<br>Select $P \in \mathcal{E}(\mathbf{F})$ of order $n$.<br>$Q = dP$<br>public key：($E$, $n$, $P$, $Q$)<br>private key：$d$ |
| Signature generation | Select $k$, $1 \le k < q$.<br>$r = (g^k \bmod p) \bmod q$<br>$s = k^{-1}(h(m) + dr) \bmod q$ | Select $k$, $1 \le k < n$.<br>$kP = (x_1, y_1)$ and $r = x_1 \bmod n$<br>$s = k^{-1}(h(m) + dr) \bmod n$ |
| Signature verification | $w = s^{-1} \bmod q$<br>$u_1 = h(m)w \bmod q$<br>$u_2 = rw \bmod q$<br>$v = (g^{u_1} y^{u_2} \bmod p) \bmod q$<br>If $v = r$, (r,s) passes | $w = s^{-1} \bmod n$<br>$u_1 = h(m)w \bmod n$<br>$u_2 = rw \bmod n$<br>$u_1P + u_2Q = (x_0, y_0)$, $v = x_0 \bmod n$<br>If $v = r$, (r,s) passes |

## Conclusions

- ECC Advantages：
  - Offers the highest strength-per-key-bit of any known public-key system
  - Offers the same level of security with smaller key sizes
  - Computational power
  - High-speed S/W and H/W implementations
  - Integrated circuit space is limited
    - smart card, wireless devices

## References

- ECC Whitepapers
  http://www.certicom.com/research/weccrypt.html
- Elliptic Curves and Cryptography

  *Dr. Dobb's Journal April 1997*

- The Elliptic Curve Digital Signature Algorithm (ECDSA)
  http://www.certicom.com/pdfs/whitepapers/ecdsa.pdf
- ECC Tutorial
  http://www.certicom.com/resources/ecc_tutorial/ecc_tutorial.html