Cryptanalysis

Elliptic Curves and Lattices

John Manferdelli
JohnManferdelli@hotmail.com

This material is provided without warranty of any kind including, without limitation, warranty of non-infringement or suitability for any purpose. This material is not guaranteed to be error free and is intended for instructional use only.

Elliptic Curves

- Motivation:
  - Full employment act for mathematicians
  - Elliptic curves over finite fields have an arithmetic operation
  - Index calculus doesn’t work on elliptic curves.
  - Even for large elliptic curves, field size is relatively modest so arithmetic is faster
  - .
- Use this operation to define a discrete log problem.
- To do this we need to:
  - Define point addition and multiplication on an elliptic curve
  - Find an elliptic curve whose arithmetic gives rise to large finite groups with elements of high order
  - Figure out how to embed a message in a point multiplication.
  - Figure out how to pick “good” curves.

Rational Points

- Bezout
- Linear equations
- $x^2+5y^2=1$
- $y^2=x^3-ax-b$
  - Disconnected: $y^2= 4x^3-4x +1$
  - Connected: a= 7, b=-10
  - Troublesome: a=3, b=-2
- Arithmetic
- $D= 4a^3-27b^2$
- Genus, rational point for g>1
- Mordell
- $Z_{n[1]}xZ_{n[2]}$, n[2]|n[1], n[2]|(p-1)

Equation solving in the rational numbers

- Linear case: Solve ax+by=c or, find the rational points on the curve C: f(x,y)= ax+by-c=0.
  - Clearing the fractions in x and y, this is equivalent to solving the equation in the integers. Suppose (a,b)=d, there are $(x, y) \in \mathbb{Z}$: ax+by=d. If d|c, say c=d'd, a(d'x)+b(d'y)=d'd=c and we have a solution. If d does not divide c, there isn't any. We can homogenize the equation to get ax+by=cz and extend this procedure, here, because of z, there is always a solution.
- Quadratic (conic) case: solve $x^2+5y^2=1$ or find the rational points on the curve C: $g(x,y)= x^2+5y^2-1=0$.
  - $(-1{,}0) \in C$. Let (x,y) be another rational point and join the two by a line: y= m(x+1). Note m is rational. Then $x^2+5(m(x+1))^2=1$ and $(5m^2+1)\ x^2+2\ (5m^2)x+(5m^2-1)= 0 \rightarrow x^2+2\ [(5m^2)/(5m^2+1)]\ x + [(5m^2-1)/(5m^2+1)]= 0$. Completing the square and simplifying we get $(x+(5m^2)/(5m^2+1))^2= [25m^4 -(25m^4 -1)]/(5m^2+1)^2= 1/(5m^2+1)^2$. So, $x= \pm(1-5m^2)/(5m^2+1)$ and substituting in the linear equation, $y= \pm(2m)/(5m^2+1)$. These are all the solutions.
- Cubic case is more interesting!

Bezout's Theorem

- Let be $f, g \ \in \ \mathbb{C}[x, y, z]$ be homogeneous polynomials with deg(f(x,y,z))=m and deg(g(x,y,z))=n . Let C_1 and C_2 be the curves in $\mathbb{C}P^2$, defined by:
  - C_1 = {(x,y,z): f(x,y,z)=0}; and,
  - C_2 = {(x,y,z): g(x,y,z)=0}.
- If f and g have no common components and $D=C_1 \cap C_2$, then $\sum_{x \epsilon D} I(C_1 \cap C_2, x)=mn$.

- I is the intersection multiplicity. This is a fancy way of saying that (multiple points aside), there are mn points of intersection between C_1 and C_2. There is a nice proof in Silverman and Tate, Rational Points on Elliptic Curves, pp 242-251. The entire book is a must read.
- A consequence of this theorem is that two cubic curves intersect in nine points.

Elliptic Curve Preliminaries -1

- Let K be a field. char(K) is the characteristic of K which is either 0 or p^n for some prime p, n>0.
- $F(x,y)= y^2+axy+by+cx^3+dx^2+ex+f$ is a general cubic.
- F(x,y) is non-singular if $F_x(x,y)$ or $F_y(x,y) \neq 0$.
- If char(K)$\neq$2,3, F(x,y)=0 is equivalent to $y^2= x^3+ax+b$ which is denoted by $E_K(a, b)$ and is called the Weierstrass equation.
- Note that the intersection of a line (y=mx+d) and a cubic, $E_K(a,b)$ is 1, 2 or 3 points.
- Idea is: given 2 points, P,Q on a cubic, the line between P and Q generally identifies a third point on the cubic, R.
- Two identical points on a cubic generally identify another point which is the intersection of the tangent line to the cubic at the given point with the cubic.
- The last observation is the motivation for defining a binary operation (addition) on points of a cubic.

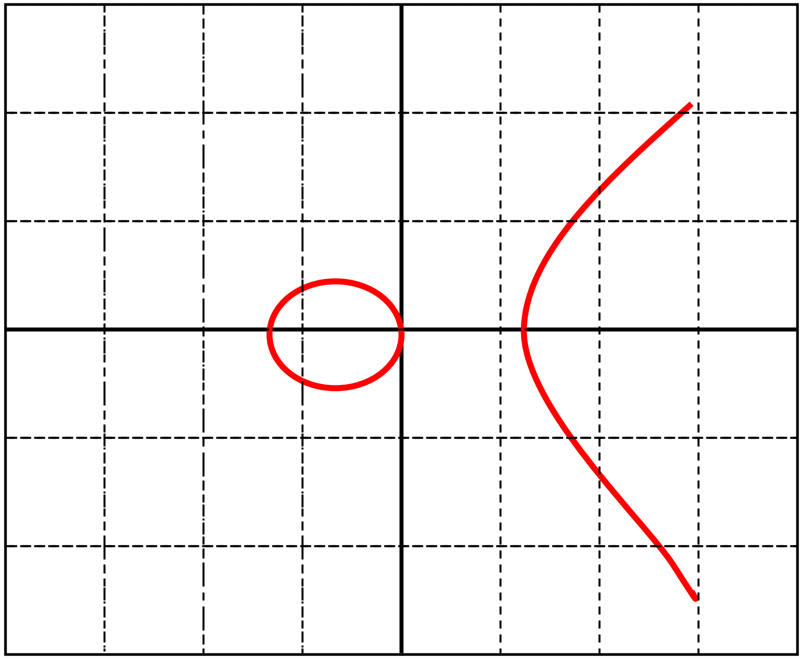
Elliptic Curve Preliminaries - 2

- We are most interested in cubics with a finite number of points.
- Cubics over finite fields have a finite number of points (duh).
- $E_K(a,b)$ is an elliptic equation over the "affine plane."

- It is often easier to work with elliptic equations over the "projective plane". The projective plane consists of the points (a,b,c) (not all 0) and (a,b,c) and (ad,bd,cd) represent the same point.
  - The map $(x,y,1) \rightarrow (xz,yz,z)$ sets up a 1-1 correspondence between the affine plane (plus the "infinities") and the projective plane.
  - $E_K(a,b)$ is $zy^2= x^3+axz^2+bz^3$. Note these are homogeneous equations.
  - The points (x,y,0) are called the line at infinity.
  - The point at infinity, (0,1,0) is the natural "identity element" O and its introduction is less "ad hoc."

Elliptic Curves

- A non-singular Elliptic Curve is a curve, having no multiple roots, satisfying the equation: $y^2=x^3+ax+b$.
  - The points of interest on the curve are those with rational coordinates which can be combined using the "addition" operation. These are called "rational points."



Graphic by Richard Spillman

Multiple roots

- Here is the condition that the elliptic curve, E_R(a, b): $y^2=x^3+ax+b$, does not have multiple roots.

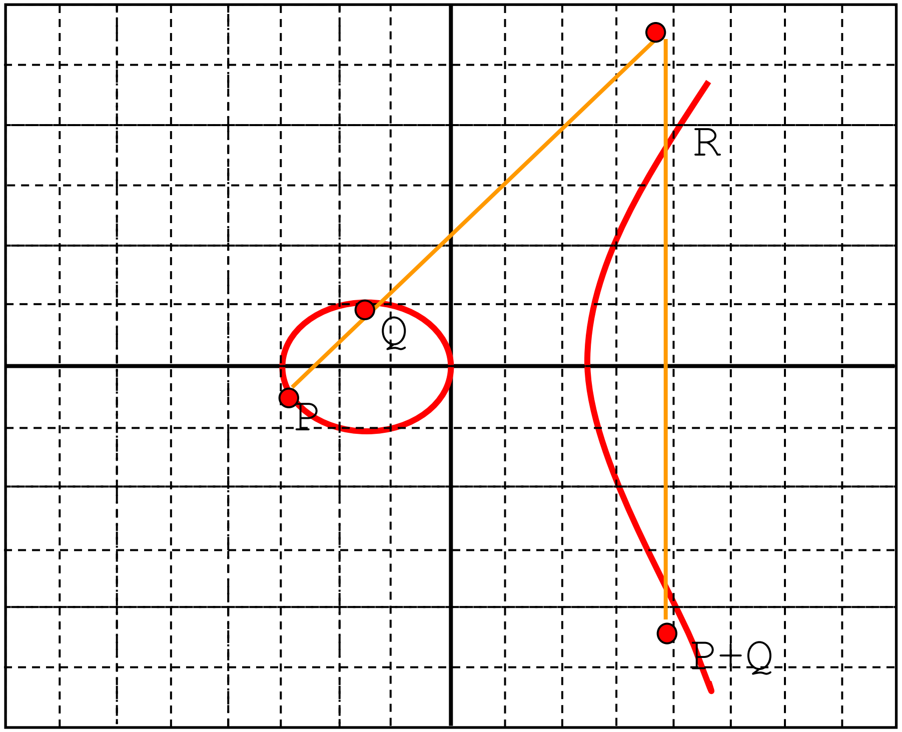
- Set $f(x,y)= y^2-x^3-ax-b=0$.
  - At a double point, $f_x(x,y)=f_y(x,y)=0$; so $f_x(x,y)= -(3x^2+a)$, $f_y(x,y)=2y$. Thus $y=0=x^3+ax+b$ and $0=(3x^2+a)$ have a common zero.
  - Substituting $a= -3x^2$, we get $0=x^3-3x^3+b$, $b= 2x^3$, $b^2=4x^6$. Cubing, $a= -3x^2$, we get $a^3= -27x^6$. So $b^2/4=a^3/(-27)$ or $27b^2+4a^3=0$. Thus, if $27b^2 + 4a^3 \neq 0$, then E_R(a, b) does not have multiple roots.

- We define the “discriminant” as $D= -16(27b^2+4a^3)$.

Elliptic curve addition

- The addition operator on a non-singular elliptic curve maps two points, P and Q, into a third "P+Q". Here's how we construct "P+Q" when P≠Q .
- Construct straight line through P and Q which hits E at R.

- P+Q is the point which is the reflection of R across the x-axis.



Graphic by Richard Spillman

Addition for points P, Q in $E_R(a, b)$ - 1

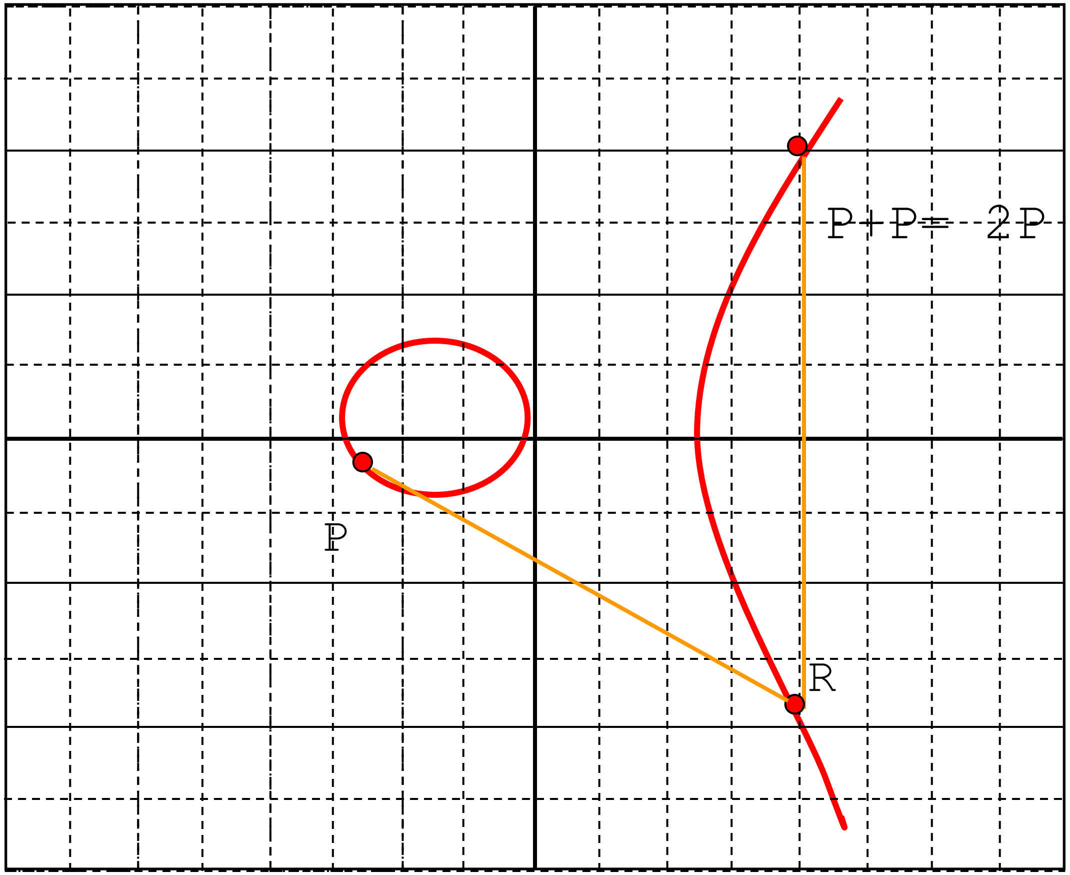
- Suppose we want to add two distinct points P and Q lying on the curve $E_R(a, b)$: $y^2=x^3+ax+b$, where $P=(x_1, y_1)$ and $Q=(x_2, y_2)$ with $P\neq Q$, then $P+Q=R=(x_3, y_3)$.

- Suppose $x_1\neq x_2$, here is the computation: Join P and Q by the line $y=mx+u$. $m=(y_2-y_1)/(x_2-x_1)$. $u= (mx_1-y_1)= (mx_2-y_2)$.
  - Substituting for y into $E_R(a, b)$, we get $(mx+u)^2=x^3+ax+b$
  - $0= x^3-m^2x+(a-2mu)x+b-u^2$.
  - x_1, x_2, x_3 are the roots of this equations so $m^2= x_1+x_2+x_3$.
  - $x_3= m^2-x_1-x_2$. $P*Q= (x_3, -y_3)$
  - $-y_3= -mx_3–u= -m(x_3) -(mx_1-y_1)= m(x_1–x_3)–y_1$.

- To summarize, if $P\neq Q$ (and $x_1\neq x_2$):
  - $x_3= m^2–x_1–x_2$
  - $y_3= m(x_1–x_3)–y_1$
  - $m= (y_2-y_1)/(x_2-x_1)$

Multiples in Elliptic Curves 1

- P+P (or 2P) is defined in terms of the tangent to the cubic at P.
- Construct tangent to P and reflect the point in y at which it intercepts the curve (R) to obtain 2P.
- P can be added to itself k times resulting in a point Q = kP.



Graphic by Richard Spillman

Addition for points P, Q in E_R(a, b) - 2

- Suppose we want to add two distinct points P and Q lying on the curve E_R(a, b): $y^2=x^3+ax+b$, where $P=(x_1, y_1)$ and $Q=(x_2, y_2)$ and $x_1=x_2$.
- Case 1, $y_1 \neq y_2$: In this case, $y_1=-y_2$ and the line between P and Q “meet at infinity,” this is the point we called O and we get P+Q=O. Note Q=-P so –(x,y)=(x,-y).
- Case 2, $y_1=y_2$ so P=Q: The slope of the tangent line to E_R(a, b) at (x_1, y_1) is m. Differentiating $y^2=x^3+ax+b$, we get $2y\ y'= 3x_2+a$, so $m=(3x_1{}^2+a)/(2y_1)$. The addition formulas on the previous page still hold.

Addition in $E_R(a, b)$ - summary

- Given two points P and Q lying on the curve $E_R(a, b)$: $y^2=x^3+ax+b$, where $P=(x_1, y_1)$ and $Q=(x_2, y_2)$ with $P \neq Q$, then $P+Q=R=(x_3, y_3)$ where:

- If $x_1 \neq x_2$, $m=(y_2-y_1)/(x_2-x_1)$, and
    - $x_3 = m^2 - x_1 - x_2$
    - $y_3 = m(x_1 - x_3) - y_1$
- If $x_1=x_2$ and $y_1 \neq y_2$, then $y_1=-y_2$ and $P+Q=O$, $Q= -P$
- If $x_1=x_2$ and $y_1=y_2$, then $P=Q$, $R=2P$, $m=(3x_1^2+a)/(2y_1)$, and
    - $x_3 = m^2 - x_1 - x_2$
    - $y_3 = m(x_1 - x_3) - y_1$

Point multiplication in $E_R(a, b)$

- By using the doubling operation just defined, we can easily calculate P, 2P, 4P, 8P ,…, 2^eP and by adding appropriate multiples calculate nP for any n.
- If nP=O, and n is the smallest positive integer with this property, we say P has order n.
- Example:
  - The order of P=(2,3) on E_R(0,1) is 6.
  - 2P=(0,1), 4P= (0,-1), 6P=O.

Example of Addition and Element Order

- E(-36,0): $y^2=x^3-36x$. P=(-3, 9), Q=(-2,8).
- P + Q = $(\lambda^2-x_1-x_2, \lambda(x_1-x_3)-y_1)$

$$\lambda = \frac{y_2-y_1}{x_2-x_1}, P \neq Q.$$

$$\lambda = \frac{3{x_1}^2+a}{2y_1}, P = Q.$$

- P+Q= (x_3,y_3)=(6,0)
- 2P=(25/4,-35/8)
- Note growth of denominators

Proof of group laws

- From the formulas and definitions, it is easy to see the operation "+" is commutative, O acts like an identity and if P= (x,y), -P= (x,-y) with P+(-P)= O.

- Associativity is the only law that's hard to verify. We could use the formulas to prove it but that's pretty ugly.
  - There is a shorter poof that uses the following result: Let C, C_1, C_2 be three cubic curves. Suppose C goes through eight of the nine intersection points of $C_1 \cap C_2$, then C also goes through the ninth intersection point.

Associativity

- If P and Q are points on an elliptic curve, E, let P*Q denote the third point of intersection of the line PQ and E.
- Now let P, Q, R be points on an elliptic curve E. We want to prove (P+Q)+R=P+(Q+R). To get (P+Q), form P*Q and find the intersection point, between P*Q and E and the vertical line through P*Q; this latter operation is the same as finding the intersection of P*Q, O (the point at infinity) and E. To get (P+Q)+R, find (P+Q)*R and the vertical line, the other intersection point with E is (P+Q)+R. A similar calculation applies to P+(Q+R) and it suffices to show (P+Q)*R=P*(Q+R). O,P,Q,R, P*Q, P+Q, Q*R, Q+R and the intersection of the line between (P+Q), R and E lie on the two cubics:
  - C_1: Product of the lines [(P,Q), (R,P+Q), (Q+R, O)]
  - C_2: Product of the lines [(P,Q+R), (P+Q,O), (R,Q)]
- The original curve E goes through eight of these points, so it must go through the ninth [(P+Q)*R]. Thus the intersection of the two lines lies on E and (P+Q)*R= P*(Q+R).
- This proof will seem more natural if you've taken projective geometry. You could just slog out the algebra though.

Mordell and Mazur

- Mordell: Let E be the elliptic curve given by the equation E: $y^2=x^3 + ax^2 + bx +c$ and suppose that $\mathrm{D}(E) = -4a^3c + a^2b^2 - 4b^3 - 27c^2 + 18abc \neq 0$. There exist r points P_1, P_2, …, P_r such that all rational points on E are of the form $a_1P_1+ \ldots +a_rP_r$ where $a_i \in \mathbb{Z}$.

- Mazur: Let C be a non-singular rational cubic curve and C(Q) contain a point of order m, then $1<m\leqq 10$ or m=12. In fact, the order of the group of finite order points is either cyclic or a product of a group of order 2 with a cyclic group of order less than or equal to 4.

Fermat's Last Theorem

- $x^n+y^n = z^n$ has no non-trivial solutions in Z for n>2.
- It is sufficient to prove this for n=p, where p is an odd prime.

- Proof (full version will be on HW):
  1. Suppose $A^p+B^p=C^p$, (A,B,C)=1.
  2. E_{AB}: $y^2 = x(x+A^p)(x+B^p)$
  3. Wiles: E_{AB} is modular.
  4. Ribet: E_{AB} is too weird to be modular.
  5. Fermat was right.

Why elliptic curves might be valuable in crypto

- Consider E: $y^2= x^3+17$. Let $P_n=(A_n/B_n, C_n/D_n)$ be a rational point on E. Define $ht(P_n)= max(|A_n|, |B_n|)$.
- Define $P_1= (2,3)$, $P_2= (-1,4)$ and $P_{n+1}= P_n + P_1$.

n	$ht(P_n)$
1	2
2	1
3	4
4	2
5	4
6	106
7	2228

n	$ht(P_n)$
8	76271
9	9776276
10	3497742218
20	8309471981636130322638066614339972215969861310

- In fact, $ht(P_n)≅(1.574)^{ns}$, $ns=n^2$.

Example from Silverman, A Friendly Introduction to Number Theory.

Points on elliptic curves over F_q

- The number of points N on $E_q(a,b)$ is the number of solutions of $y^2=x^3+ax+b$.
- For each of q x's there are up to 2 square roots plus O, giving a maximum of 2q+1. However, not every number in F_q has a square root. In fact, N= $q+1+\sum_x\chi(x^3+ax+b)$, where χ is the quadratic character of F_q.
- *Hasse's Theorem:* $|N-(q+1)|\leq 2\sqrt{q}$ where N is the number of points
- $E_q(a,b)$ is supersingular if N= (q+1)-t, t= 0,q, 2q, 3q or 4q.
- The abelian group formed by addition in $E_q(a,b)$ does not need to be cyclic, although it often is; it can always be decomposed into cyclic groups. In fact, if G is the Elliptic group for $E_q(a,b)$.
- *Theorem:* $G=Z_p \times Z/Zp^a \times Z/Zq^b$.
- Example: $E_{71}(-1,0)$. N= 72, G is of type (2,4,9).

$E_{71}(-1, 0)$ – Spot the Group

- There are 72 points on the curve. Can you spot (2, 4, 9). Points:

Order	Point	Order	Point	Order	Point	Order	Point
[1]	O	[18]	(14, 48)	[12]	(40, 29)	[18]	(53, 24)
[2]	(0, 0)	[3]	(19, 38)	[36]	(41, 62)	[36]	(54, 28)
[2]	(1, 0)	[3]	(19, 33)	[36]	(41, 9)	[36]	(54, 43)
[9]	(2, 19)	[36]	(21, 62)	[18]	(42, 8)	[12]	(55, 31)
[9]	(2, 52)	[36]	(21, 9)	[18]	(42, 63)	[12]	(55, 40)
[18]	(3, 38)	[18]	(23, 28)	[36]	(43, 21)	[6]	(56, 41)
[18]	(3, 33)	[18]	(23, 43)	[36]	(43, 50)	[6]	(56, 30)
[9]	(4, 42)	[36]	(27, 42)	[36]	(45, 49)	[4]	(60, 10)
[9]	(4, 29)	[36]	(27, 29)	[36]	(45, 22)	[4]	(60, 61)
[18]	(5, 7)	[12]	(32, 54)	[36]	(46, 37)	[36]	(61, 2)
[18]	(5, 64)	[12]	(32, 17)	[36]	(46, 34)	[36]	(61, 69)
[6]	(9, 62)	[36]	(33, 7)	[18]	(47, 51)	[6]	(63, 8)
[6]	(9, 9)	[36]	(33, 64)	[18]	(47, 20)	[6]	(63, 63)
[36]	(12, 56)	[18]	(35, 58)	[18]	(49, 38)	[36]	(64, 27)
[36]	(12, 15)	[18]	(35, 13)	[18]	(49, 33)	[36]	(64, 44)
[4]	(13, 14)	[9]	(37, 8)	[12]	(51, 16)	[36]	(65, 28)
[4]	(13, 57)	[9]	(37, 63)	[12]	(51, 55)	[36]	(65, 43)
[18]	(14, 23)	[12]	(40, 42)	[18]	(53, 47)	[2]	(70, 0)

Addition for points P, Q in $E_p(a, b)$

1. P+O=P
2. If P=(x, y), then P+(x, -y)=O. The point (x, -y) is the negative of P, denoted as –P.
3. If $P=(x_1, y_1)$ and $Q=(x_2, y_2)$ with P≠Q, then $P+Q=(x_3, y_3)$ is determined by the following rules:
   - $x_3 = \lambda^2 – x_1 – x_2$ (mod p)
   - $y_3 = \lambda(x_1 – x_3) – y_1$ (mod p)
   - $\lambda = (y_2 - y_1)/(x_2 - x_1)$ (mod p) if P≠Q
   - $\lambda = (3(x_1)^2 + a)/(2y_1)$ (mod p) if P=Q
4. The order of P is the smallest positive number n: nP=O

Point multiplication in $E_p(a, b)$

- E: $y^2 = x^3+17 \pmod{101}$ or $E_{101}(0,17)$
  - $x_3 = m^2 - x_1 - x_2 \pmod{p}$
  - $y_3 = m(x_1 - x_3) - y_1 \pmod{p}$
  - $m=(y_2-y_1)/(x_2-x_1) \pmod{p}$ if $P \neq Q$
  - $m=(3(x_1)^2+a)/(2y_1) \pmod{p}$ if $P=Q$
- (23,93)+(54,74)= (29, 41)
  - m= (74-93)/(54-23) = -19/31= 82 x 88 = 45
  - $x_3 = 45^2$-23-54= 29 (101)
  - y_3 =45 x (23-29)-93)= 41
- 2 x (41, 37)= (35, 88)
  - m= (3 x 41^2 + 0)/(2 x 37)= 94/74=94 x 86 =4
  - $x_3 = 4^2$- 82= 35
  - y_3 =4 x (41-35)-37= -13= 88 (101)

Note:

$93^2= 23^3+17=64$ (101)

$74^2= 54^3+17=22$ (101)

$41^2= 29^3+17=65$ (101)

$37^2= 41^3+17=56$ (101)

$88^2= 35^3+17=64$ (101)

Elliptic Curve (Characteristic = 2)

- For K of characteristic 2, define $j(E) = (a_1)^{1/2}/\Delta$
- If $j(E) \neq 0$:
  - $-P = (x_1, y_1+x_1)$
  - $P+Q = (x_3, y_3)$
  - $P \neq Q$
    - $x_3 = ((y_1+y_2)/(x_1+x_2))^2+(y_1+y_2)/(x_1+x_2) +x_1+x_2+a$,
    - $y_3 = ((y_1+y_2)/(x_1+x_2))(x_1+x_3) + x_3 + y_1$
  - $P = Q$
    - $x_3 = x_1^2+b/x_1^2$,
    - $y_3 = x_1^2+(x_1+y_1/x_1)x_3+x_3$

If $j(E) = 0$:

- $-P = (x_1, y_1+c)$
- $P+Q = (x_3, y_3)$
- $P \neq Q$

  $x_3= ((y_1+y_2)/(x_1+x_2))^2 +x_1+x_2$

  $y_3 = ((y_1+y_2)/(x_1+x_2))(x_1+x_3)+c+ y_1$
- $P = Q$

  $x_3 = (x_1^4+a^2)/ c^2$, $P = Q$

  $y_3 = ((x_1^2+a)/c)(x_1+x_3)+c+y_1$

Structure of the Elliptic Curve Group on $E_p(a,b)$ - 1

- $E_{11}(1, 6)[\ y^2= x^3 + 1\ x + 6 \pmod{11}]$. D: -7, 2 is primitive (mod 11). $D=4a^3+27b^2 \pmod{p}$. 13 points on curve; G, cyclic.

Order	Point
[1]	O
[13]	(2, 4)
[13]	(2, 7)
[13]	(3, 5)
[13]	(3, 6)
[13]	(5, 2)
[13]	(5, 9)
[13]	(7, 2)
[13]	(7, 9)
[13]	(8, 8)
[13]	(8, 3)
[13]	(10, 2)
[13]	(10, 9)

Powers	
(1)	(5, 2)
(2)	(10, -9)
(3)	(7, 9)
(4)	(3, 5)
(5)	(8, 8)
(6)	(2, 4)
(7)	(2, 7)
(8)	(8, 3)
(9)	(3, 6)
(10)	(7, 2)
(11)	(10, 9)
(12)	(5, 9)
(13)	O

Structure of the Elliptic Curve Group on $E_p(a,b)$ - 2

- $E_{31}(1, 6)$. D: -23, 3 is primitive (31). 32 points on curve. Not cyclic!

Order	Point	Order	Point
[1]	O	[16]	(19, 8)
[16]	(1, 16)	[16]	(19, 23)
[16]	(1, 15)	[4]	(20, 20)
[8]	(2, 27)	[4]	(20, 11)
[8]	(2, 4)	[16]	(21, 9)
[4]	(3, 25)	[16]	(21, 22)
[4]	(3, 6)	[16]	(24, 20)
[2]	(9, 0)	[16]	(24, 11)
[16]	(12, 17)	[16]	(25, 30)
[16]	(12, 14)	[16]	(25, 1)
[8]	(14, 25)	[2]	(26, 0)
[8]	(14, 6)	[2]	(27, 0)
[16]	(17, 10)	[8]	(28, 10)
[16]	(17, 21)	[8]	(28, 21)
[16]	(18, 20)	[8]	(30, 29)
[16]	(18, 11)	[8]	(30, 2)

Structure of the Elliptic Curve Group on $E_p(a,b)$ - 3

$E_p(a, b)$ $y^2= x^3 + ax + b$ (mod p). D= $4a^3+27b^2$ (mod p).

Cyclic

$E_{29}(0, 17)$. D: -3, <2> (29). 30 points. G: (2,24).
$E_{31}(0, 17)$. D: -11. <3> (31). 43 points. G: (1, 24).
$E_{101}(0, 17)$. D: -12. <2> (101). 102, points. G: (4, 9).
$E_{311}(0, 17)$. D: -137. <17> (311). 312 points. G: (14, 133).
$E_{29}(1, 6)$. D: -14. <2> (29). 38 points. G: (2, 4).
$E_{47}(1, 6)$. D: -12. <5> (47). 52 points. G: (0, 10).
$E_{101}(1, 6)$. D: -62. <2> (101). 112 points. G: (0, 39).
$E_{1217}(0, 17)$. D: -714. <3> (1217). 1218 points. G: (2, 5).

Not cyclic

$E_{31}(1, 6)$. D: -23. <3>(31). 32 points. (1, 6) has order 16.

Group order and Hasse

- $\#E_q(a,b)= q+1-t$
  - $j^2- [t]j+q=0$
  - $|t| \leqq \sqrt{2q}$
- $G(E_p(a,b)) = Z_n \times Z_m$, $n|m$, $n|p-1$. Used proving endomorphisms.
- Let E be an elliptic curve over K and n a positive integer. If char(K) does not divided n or is 0, then $E[n]= Z_n \times Z_n$.
- Twist: m: $a_2= m^2a_1$, $b_2= m^3b_1$.
  - $\#E_p(a_1,b_1)+\#E_p(a_2,b_2) = p+2$

Point counting

- Group order calculations are critical for curve selection and algorithm safety. The number of points on the curve is the size of the group so counting points is important. There are several methods:
  1. Baby Step Giant Step: Explained in next slide.
  2. Schoof: O($\lg^8$(p)). Beyond the scope of this lecture.
     Determines t (mod l) for l, prime and $I \leq I_{max}$, where P_l l $>4\sqrt{p}$.
  3. SEA: Schoof-Elkies-Atkins. Further beyond the scope of this lecture.

Elliptic Curve Discrete Log Problem

- Let C be an elliptic curve, E(a,b): $y^2=x^3+ax+b$, over a finite field K with elliptic group, G. Given P, Q in the group with P=nQ, find n.
- Elliptic Curve crypto system is precisely analogous to discrete log systems using arithmetic over finite fields.
  - Discovered by Koblitz and Miller
- Note in computing kP over $E_p(a,b)$, we can write k as powers of 2 and multiply P by k in $\lg(k)\lg(p)^3$ time. For example, $40P= (2^5+2^3)P$

Baby step, giant step

- Want to find m: O= [m]P. There is a general attack just like in DLP called the Baby Step – Giant Step Attack. It takes O(√n) where n is the order of the group.
- The attack:
  1. M=ceiling(√n). m=aM+b is the order of P.
  2. To find a, b note (O-[b]P)=[a][M]P.
  3. Compute R_b= O-[b]P, b=1,2,…,M. Store (b, O-[b]P) sorted by second element.
  4. Giant step: S_a= [a][M]P, a= 1,2,…,M check table if
  5. S_a=R_b, m=aM+b.

Special Attacks on discrete log in $E_q(a,b)$

- MOV Attack (Menezes, Okamoto, Vanstone).
  - Idea: map the ECDLP to the DLP in an extension field.
- In the case of MOV, if n is the order of a point (hence it divides the number of points on the curve) and $n|q^k-1$, the ECDLP can be mapped into the DLP in $GF(q^k)$.
  - To avoid this attack, we need to make sure the DLP in $GF(q^l)$ is as hard as the ECDLP in $E_q(a,b)$. This is guaranteed to happen of $l>k^2/(lg(k)^2)$, so we can avoid this attack if the smallest l: $q^l=1$ (mod n) satisfies $l>k^2/(lg(k)^2)$.
- Another attack: An anomalous curve satisfies $\#E_q(a,b)=q$. This group is cyclic and allows an easy embedding in the DLP problem in the additive group of F_q. To avoid this, make sure the number of points on the elliptic curve is not q.

Diffie Hellman over *ECC*

- Alice and Bob chose a finite field F_q and an elliptic curve E
- The key will be taken from a random point P over the elliptic curve (e.g. - the x coordinate).
- Alice and Bob choose a point B that does not need to be secret
  - B must have a very large order
- Alice chooses a random *a* and compute $aB \in E$
- Bob chooses a random *b* and compute $bB \in E$
- Alice and Bob exchange the computed values

- Alice, from bB and a can compute P= abB
- Bob, from aB and b can compute P= abB

Elliptic curve El Gamal

- There are several ways in which the ECDLP can be embedded in a cipher system.
  - One method begins by selecting an Elliptic Curve, $E_p(a,b)$, a point G on the curve and a secret number k which will be the private key.
  - The public key is G and P_A where P_A= kG. Think of G as the generator in the discrete log problem.
  - A message is encrypted by converting the plaintext into a number m, selecting a random number r, and finding a point on the curve P_m corresponding to m. We explain how to do this in the next slide.
  - The ciphertext consists of two points on the curve $\{rG, P_m+r\ P_A\}$
  - To decipher, multiply the first point by k and subtract the result from the second point: $P_m+rP_A–k(rG)= P_m+r(kG)–k(rG)= P_m$.

Embedding m in $E_q(a,b)$

- There is no deterministic way.
- Assume $q= p^r$ and we want to embed with a probability of failure not to exceed 2^{-k}.
- Message is m and $0 \leq m < M$. $q > M\kappa$.
- For $a^{r-1}p^{r-1}+\ldots+a_1p+a_0=x_a= m\kappa+j$.
- For j= 0, try to solve $y^2=x_a^3+ax_a+b$ by evaluating Legendre symbol. Can do this with probability ½. If this succeeds, use it. Otherwise try j=1 ,…
- Given x_a, we can recover m by writing $x_a=m\kappa +j$ and discarding j.
- $P_m= (x_a, y)$.

Putting it all together: EC El Gamal

- Curve: $E_{8831}(3,45)$
- G=(4,11),a=3, A=aG=(413,1808)
- b=8, B=bG= (5415, 6321)
- P= (5, 1743)
- Bob sends Alice:
  - [B, P+ 8A]= [(5415,6321), (6626,3576)]
- Alice decrypts as:
  - 3 (5415, 6321)= (673, 146)
  - P= (6626,3576)-(673,146)= (6626,3576)+(673,-146)= (5, 1743)

Putting it all together: ECDH

- Curve: E_{7311}(1,7206)
- G=(3,5)
- Alice picks a=12 sends aG= (1794,6375)
- Bob picks b= 23, sends bG= (3861,1242)
- Bob computes 23(1794, 6375)= (1472, 2098)
- Alice computes 12 (3861,1242)= (1472, 2098)

Picking Curves

- Curves are selected at random subject to resistance to known attacks like Hellman-Pohlig-Silver and Pollard rho.
    1. $\#E(F_q)$ should be divisible by a large prime, n.
    2. $\#E(F_q)$ should not be q
    3. n should not divide q^k-1
- Method of selecting curves
    - Select a,b at random with $(4a^3+27b^2)\neq 0$
    - Calculate N= $\#E(F_q)$.
    - Factor N and verify 1, 2, 3 above.
    - If the coefficients are selected at random, the order of the curves are uniformly distributed (Lenstra).

Curve selection

- Given p and a parameter S, generate an acceptable E.

1. Generate random a,b**ε**F_p.
2. If Δ=0 go to 1.
3. Determine N= #E_p(a,b)
4. If E_p(a,b) is anomolous (p=N), go to 1.
5. If E_p(a,b) is subject to MOV attack, there is an $l<lg(p)^2/(lg(lg(p))^2$: $p^l=1$ (mod N), go to 1.
6. Factor N, if it takes too long, go to 1.
7. If N=sxr, s≦S return E_p(a,b)
8. Go to 1.

ECC Point Operation Costs and modular operations

Parameters

- I= inverse cost in GF(p).
- S= square cost GF(p).
- M= multiply cost GF(p)

Op	**Cost**	**Modular Op**	**Cost**
2P	I+2S+2M	Add, Sub	O(lg(n))
P+Q	I + S+ 2M	Multiply	O(lg(n)^2)
2P+Q	2I + 2S + 2M	Invert	O(lg(n)^2)
P+Q, P-Q	I+2S+4M	Exp	O(lg(n)^3)

ECC vs RSA performance analysis

- n= [lg(p)] (for EC), N= [lg(p)] for DLP.
- The cost to break DLP with best known algorithm (IC) is $c_{DLP}(N)= \exp(c_0 N^{1/3} \ln(N \ln(2))^{2/3})$.
- The cost to break ECDLP with best known algorithm (IC) is $c_{ECDLP}(n)= 2^{n/2}$.
- $n= b(N^{1/3}) \ln(N(\ln(2))^{2/3}$, $b=2c_0/\ln(2^{)2/3}$~4.91
- The number of key bits (for equivalent security) in the DLP case grows as the cube of the number of bits for the ECDLP case. This has a key size and performance implication.

Pollard Rho Method for ECC vs. Factoring by Number Field Sieve

Key size	MIPS-Years
150 bits	$3.8\text{x}10^{10}$
205 bits	$7.1\text{x}10^{18}$
234 bits	$1.6\text{x}10^{28}$

- Elliptic Curve Logarithms Using Pollard Rho Method

Key size	MIPS-Years
512 bits	$3\text{x}10^{4}$
768 bits	$2\text{x}10^{8}$
1024 bits	$3\text{x}10^{11}$
1280 bits	$3\text{x}10^{14}$
1536 bits	$3\text{x}10^{16}$
2048 bits	$3\text{x}10^{20}$

- Integer Factoring Using Number Field Sieve

This slide came from someone else

Observations on ECC

- Asymmetry between encryption and decryption is reduced (4:1)
- NIST recommendations for key size to provide “equivalent” security (bits in key).

ECC	RSA	AES
163	1024	
256	3072	128
384	7680	192
521	15360	256

NIST Curves

- Use prime fields F_p with $p=2^{192}-2^{64}-1$, $p=2^{224}-2^{96}+1$, $p=2^{256}-2^{224}+2^{192}+2^{96}-1$, $p= 2^{384}-2^{128}- 2^{96}+2^{32}-1$, $p=2^{521}-1$ or binary fields F_q with $q= 2^{163}$, 2^{233}, 2^{283}, 2^{409}, 2^{571}.
- $\#E_p(a,b)=q+1-t$, $|t|\leqq 2\sqrt{q}$ and t is called the trace of E. $E_q(a,b)$ has rank 1 or 2, that is: $E_q(a,b) \sim Z_{n[1]} x Z_{n[2]}$ and $n[2] \mid n[1]$, $n[2] \mid (q-1)$.
- If $n[2] =1$, $E_q(a,b) \sim Z_{n1}= \{kP: 0<k<n[1]\}$ and P is a generator.
- $E_q(a_1, b_1) \sim E_q (a_2, b_2)$ if $a_1= u^4a_2$ and $b_1= u^4b_2$.
- E_q, $q= p^n$ is supersingular if $p|t$. Field represented as polynomial or normal basis.

El Gamal Signature

- Bob has a private key x and a public key <g,X>: X= g^x in a group G. To sign m, given a map f: G → $Z_{|G|}$:
  1. Bob generates a random a: 1≦a<|G|. A= g^a.
  2. Bob computes B$\boldsymbol{\varepsilon}$$Z_{|G|}$: m=xf(A)+Ba (mod |G|).
  3. $\mathrm{Sig}_{\mathrm{Bob}}$(m)= (A,B)
- To verify check that the signature is right, verify that $X^{f(A)}A^B$=g^m.

EC El Gamal Signature

- Bob has a private key x and a public key <g,X>: X= g^x in a group G. To sign m, given a map f: G → $Z_{|G|}$:
  1. Bob generates a random a: $1 \leqq a < |G|$. A= g^a.
  2. Bob computes $B \boldsymbol{\varepsilon} Z_{|G|}$: m=xf(A)+Ba (mod |G|).
  3. Sig_{Bob}(m)= (A,B)
- To verify check that the signature is right, verify that $X^{f(A)}A^B=g^m$.

Factoring using Elliptic Curves

- Let $E_n(a,b)$ be an elliptic curve with $(4a^3+27b^2, n)=1$ and let P_1, P_2 be two rational points whose denominators are prime to n. Then $O \neq P_1+P_2 \boldsymbol{\varepsilon} E$ has denominators prime to n iff there is no prime p|n such that $P_1+P_2 = O$ (mod p).

- Lenstra's Algorithm. Choose 2 bounds B, K.
  1. $(n,6)=1$, $n \neq m^r$
  2. Choose random b, x_1, y_1 between 1 and n
  3. $c= {y_1}^2+ {x_1}^3-bx_1$ (mod n)
  4. $(n,4b^3+27c^2)=1$
  5. k= LCM(1,2,…,K)
  6. Compute $kP=(a_k/{d_k}^2,b_k/{d_k}^3)$, if at any point can't succeed, n is composite.
  7. $D=(d_k,n)$. If D=1, go to 5 and bump K or go to 2 and select new curve.

Factoring using elliptic curves - example

- Factor n=4453.
- Use E: $y^2 = x^3+10x-2$ (mod m).
- Initial point: P_1= (1,3).
- 2P=(4332, 3230).
- To calculate 3P:
  - m=(3230-3)/(4332-1)=3227/4331.
- (4331, 4453)=61.
- 4453= 61x73.

Factoring using elliptic curves - example

- Factor m=1938796243.
- Use E: $y^2 = x^3 - Ax + A \pmod{p}$. A= 1,2,…
- Initial point: $P_1 = (1,1)$, $P_{n+1} = (n+1)P_n$.
- For A=7, $(w_{16}, m) = 37409$. m= 37409 x 51827.
- $a_i = a^{(r_1 r_2 \dots r_i)}, g_i = (a_n - 1, n)$

Divisors

- $D= \sum_j a_j[P_j]$, a_j **ε** Z
- $\deg(D)= \sum_j a_j$
- $\mathrm{sum}(D)= \sum_j a_jP_j$
- sum: $\mathrm{Div}^0(E) \rightarrow E(K)$.
- $f= u_P{}^r g$: $\mathrm{ord}_P(f)=r$, $\mathrm{div}(f)= S_{P\boldsymbol{\varepsilon} E(K)} \mathrm{ord}_P(f)\ [P]$
- $\mathrm{Div}^0(E)$/(principal divisors) is isomorphic to E(K)
- Let E be an elliptic curve and f a function on E that is ≠0 then
  1. f has only finitely many poles and zeros
  2. deg(div(f))=0
  3. If f has no poles or zeros it is constant

Pairings

- $E[n] \subseteq E(K)$, e_n: $E[n] \times E[n] \rightarrow \mu_n$
- $T \varepsilon E[n]$, f: $\mathrm{div}(f)= n[T]-n[\infty]$
- Choose $T' \varepsilon E[n^2]$: $nT'=T$: $\mathrm{div}(g)= S_{R \varepsilon E[n]}([T'+R]-[R])$
- $\mathrm{div}(f \circ n)=\mathrm{div}(g^n)$
- Let $S \varepsilon E[n]$, $P \varepsilon E(K)$ then $g(P+S)^n= f(n(P+S))=f(nP)=g(P)^n$
  - Thus $g(P+S)/g(P)$ ε μ_n and is independent of P.
- Define $e_n(S,T)= g(P+S)/g(P)$, then
  - e_n: $E[n] \times E[n] \rightarrow \mu_n$
  - e_n is binlinear, non-degenerate.
  - $e_n(\sigma S, \sigma T)= \sigma e_n(S,T)$
  - $e_n(\alpha S, \alpha T)= e_n(S,T)^{\deg(\alpha)}$ if a is separable.

Lattices

- The set $\Lambda = \mathbb{Z}b_1 + \mathbb{Z}b_2 + \ldots + \mathbb{Z}b_n$, where $b_1, b_2, \ldots, b_n$ are linearly independent is called a lattice.
- $\Lambda^* = \{y \in \mathbb{Z}^n : (x, y) \in \mathbb{Z}, \forall x \in \Lambda\}$
- $vol(\Lambda) = \det(b_1, b_2, \ldots, b_n)$, where $b_1, b_2, \ldots, b_n$ are the generators of Λ. Note that any set of generators will do since they are related by uni-modular transformations.
- Let Λ be a lattice
  - The CVP problem is: Find $v \in \Lambda$: $\left||v|\right| = min_{w\in\Lambda, w\neq 0}(||w||)$
  - The CVP_γ problem is: Find $v \in \Lambda$: $\left||v|\right| \leq \gamma \cdot min_{w\in\Lambda, w\neq 0}(||w||)$

Definitions

- Hermite Normal Form (HNF)

$$\begin{bmatrix} >0 & 0 & 0 & \cdots & 0 & 0 & \ldots & 0 \\ \geq 0 & >0 & 0 & 0 & 0 & 0 & \ldots & 0 \\ \geq 0 & \vdots & >0 & \ddots & \vdots & 0 & \ldots & 0 \\ \geq 0 & \geq 0 & \geq 0 & \ldots & 0 & 0 & \ldots & 0 \\ \geq 0 & \geq 0 & \geq 0 & \cdots & >0 & 0 & \ldots & 0 \end{bmatrix}$$

Minkowski’s Theorem

- Let Λ be a lattice in $\mathbb{R}^n$ and suppose $S \subseteq \mathbb{R}^n$ is a convex, centrally symmetric region. If $vol(S) > 2^n \det(\Lambda)$ then S has a non-zero lattice point of Λ.

Suppose first that Λ' is the simple lattice generated by $e_1, e_2, \ldots\ e_n$. Represent a point $\mathrm{r} \in S$ as $\mathrm{r} = (\alpha_1 + x_1, \alpha_2 + x_2, \ldots, \alpha_n + x_n)$ with $\alpha_i \in \mathbb{Z}$ and $|x_i| \leq 1$, for $1 \leq i \leq n$. Define $T(r) = (x_1, x_2, \ldots, x_n)$. If $S_1 \cap S_2 = \emptyset$, $vol(S_1 \cup S_2) = vol(S_1) + vol(S_2)$. So, if S has the property that $T(t) \neq T(s), \forall s \neq t \in S$, then $vol(S) = vol(T(S))$. Note that $vol(T(S)) \leq 1$. So, if $vol(S) > 1$, there are at least two points $r^{(1)} = \left(\alpha_1{}^{(1)} + x_1, \alpha_2{}^{(1)} + x_2, \ldots, \alpha_n{}^{(1)} + x_n\right), r^{(2)} = \left(\alpha_1{}^{(2)} + x_1, \alpha_2{}^{(2)} + x_2, \ldots, \alpha_n{}^{(2)} + x_n\right)$, where $\alpha_i{}^{(1)} \neq \alpha_i{}^{(2)}$ for some i. Since S is centrally symmetric, $-r^{(1)}, -r^{(2)} \in S$; finally, note that $0 \neq r^{(1)} - r^{(2)} \in \mathbb{Z}^n$. Similarly, if $vol(S) > 2^n$, there are at least $2^n + 1$ points $r^{(i)}$, $1 \leq i \leq 2^n + 1$ with $0 \neq r^{(i)} - r^{(j)} \in \mathbb{Z}^n, i \neq j$ for at least two, say $r^{(i)}$ and $r^{(j)}$, all corresponding coordinates in $r^{(i)} - T(r^{(i)})$ and $r^{(j)} - T(r^{(j)})$ are equal $(mod\ 2)$. Thus, $0 \neq \frac{r^{(i)} - r^{(j)}}{2} \in \mathbb{Z}^n$. But since S is convex, $\frac{r^{(i)} - r^{(j)}}{2} \in S$. So, the result holds for the simple lattice. Suppose now that Λ is genetated by $a_1, a_2, \ldots\ a_n$ and put $A = [a_1, a_2, \ldots\ a_n]$. $e_i = A^{-1}(a_i)$, so $vol(\Lambda') = \frac{vol(\Lambda)}{\det(\Lambda)}$ and the simple lattice result thus implies the general theorem.

q-ary lattices and other definitions

- Definition: If $q \in \mathbb{Z}$, a lattice, Λ, is called q-ary if $q\mathbb{Z}^n \subseteq \Lambda \subseteq \mathbb{Z}^n$.
- Suppose $A \in \mathbb{Z}^{m \times n}$, $\Lambda_q(A) = \{\mathrm{y} \in \mathbb{Z}^n : y = A^T x \ (mod\ q),\ x \in {\mathbb{Z}_q}^m\}$. Note $\Lambda_q(A)$ is q-ary.
- ${\Lambda_q}^{\perp}(A) = \{\mathrm{y} \in \mathbb{Z}^n : Ay = 0 \ (\mathrm{mod\ q})\}$
- $\lambda_1(\Lambda) = \left| min_{v \in \Lambda} \, ||\mathrm{v}|| \right|$
- $\lambda_n(\Lambda) = \ min_S(max_{v \in S}\left||v|\right|)$, where $S \subseteq \Lambda$ is a set of linearly independent vectors, $|S| = n$

- Solving CVP in ${\Lambda_q}^{\perp}(A)$ when A is chosen uniformly at random is as hard as worse case CVP.

Some simple results

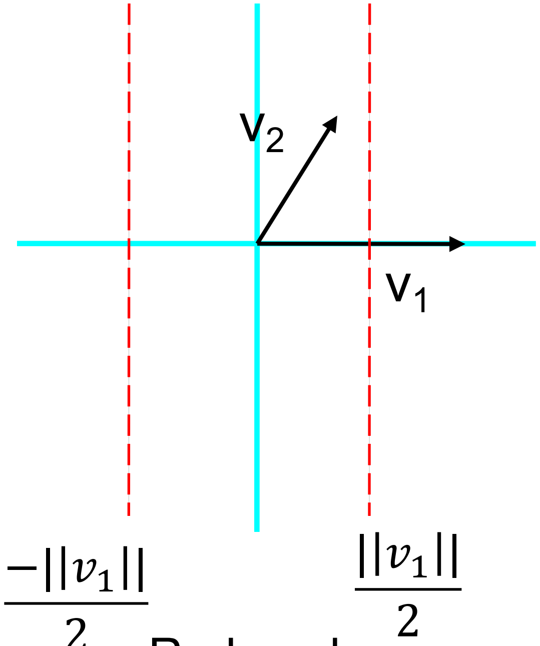
- Remember S is centrally symmetric if $s \in S$ implies $-s \in S$, and S is convex if $s, t \in S$ implies $us + (1-u)t \in S, u \in [0,1]$. We used this in proving Minkowski.

- Theorem: $\lambda_1(\Lambda) \leq \sqrt{n}\,\det(\Lambda)^{\frac{1}{n}}$

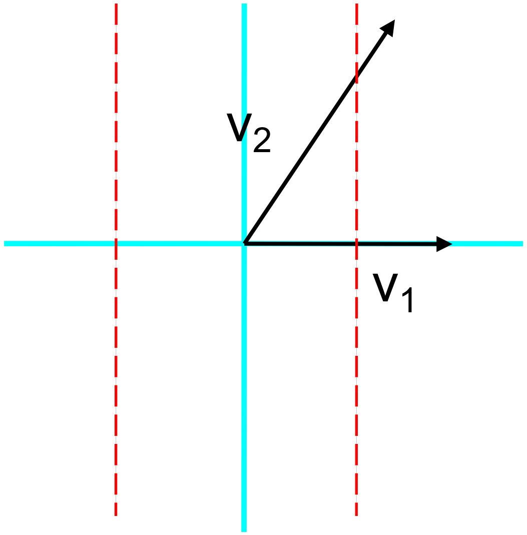
  Let B_r be a ball centered at 0 having radius $r = \sqrt{n}\,\det(\Lambda)^{\frac{1}{n}}$. Let $(x_1, x_2, \dots, x_n)$ be the coordinates of a vector v, with respect to the basis generating the lattice Λ, if $|x_i| \leq 1$ for $1 \leq i \leq n$, $v \in B_r$. So $-\det(\Lambda)^{\frac{1}{n}}\ (1,1,\dots,1)$ and $\det(\Lambda)^{\frac{1}{n}}\ (1,1,\dots,1)$ as well as the line joining them are in B_r so $vol(B_r) \geq 2^n \det(\Lambda)$ and the result follows from Minkowski's theorem.

Reduced Basis

- $<v_1, v_2>$ is reduced if
  - $||v_2|| \leqq |v_1||$; and,
  - $-1/2||v_1||^2 \leqq (v_1, v_2) \leqq 1/2||v_1||^2$.



Reduced



Not

Good basis and Gram-Schmidt Orthogonalization

- Good basis for lattices are orthonormal when that is possible. If a basis, $b_1, b_2, \ldots, b_n$ for Λ, is orthonormal, then, for example, $vol(\Lambda)$ = $||b_1|| \cdot ||b_2|| \cdot \cdots \cdot ||b_n||$
- The orthogonality defect of a basis $b_1, b_2, \ldots, b_n$ is $\frac{||b_1|| \cdot ||b_2|| \cdot \ldots \cdot ||b_n||}{\det(b_1, b_2, \ldots, b_n)}$
- Given a space generated by $b_1, b_2, \ldots, b_n$ can also be generated by a set of vectors, ${b_1}^*$, ${b_2}^*$,…, ${b_n}^*$ with the property that $({b_i}^*, {b_j}^*) = 0, i \neq j$. Th Gram-Schmidt orthogonalization procedure computes this.

```
GSO, given, $b_1, b_2, \ldots, b_n$, compute ${b_1}^*$, ${b_2}^*$,…, ${b_n}^*$
    1. put ${b_1}^* = b_i$.
    2. for $i = 2, i \leq n$
        ${b_i}^* = b_i - \sum_{j=1}^{i-1} \mu_{i,j} b_j$ , $\mu_{i,j} = \frac{({b_j}^*, b_i)}{({b_j}^*, {b_j}^*)}$
```

Size Reduction

- Definition: A basis $b_1, b_2, \ldots, b_n$ is *size reduced* if $\left|\mu_{i,j}\right| \leq \frac{1}{2}$, in the Gram-Schmidt orthogonalization procedure.
- If $b_1, b_2, \ldots, b_n$ is a basis for Λ, in general, ${b_1}^*, {b_2}^*$,…, ${b_n}^*$ is not also a lattice basis because $\mu_{i,j}$ is generally not an integer. We can find a “nearly” orthogonal set of vectors $b_1', b_2', \ldots, {b_n}'$ in Λ, by rounding the $\mu_{i,j}$. $b_1', b_2', \ldots, {b_n}'$ is also a basis for the lattice and has the same gram Schmidt basis, ${b_1}^*, {b_2}^*$,…, ${b_n}^*$. When performing GSO on this *reduced* basis, $\left|\mu_{i,j}\right| \leq \frac{1}{2}$.

Size-reduction

```
    for $i = 2, i \leq n$
        for $j = i - 1, j \geq 1$
            $b_i \leftarrow b_i - \lceil \mu_{ij} \rfloor b_j$
            for $k = 1, k \leq j$
                $\mu_{ik} \leftarrow \mu_{ik} - \lceil \mu_{ij} \rfloor \mu_{jk}$
```

Size reduction and basis reordering

- Let $b_1, b_2, \ldots, b_n$ be a basis for Λ, and $b_1{}^*, b_2{}^*$,…, $b_n{}^*$ the resulting GSO basis. Let $B_i = ||b_i||^2$. Then $b_1, b_2, \ldots, b_n$ satisfies the *Lovasz condition* with factor δ if it is size reduced and $\left(\delta - \mu_{i+1,i}{}^2\right)B_i \le B_{i+1}$. The LLL algorithm calculates such a basis.

LLL Algorithm

```
Given $b_1, b_2, \ldots, b_n$ generating $\Lambda$, calculate the LLL reduced basis
1. Reduce the basis $b_1, b_2, \ldots, b_n$ with the size reduction algorithm and calculate
   $b_1{}^*, b_2{}^*$,…, $b_n{}^*$ and $\mu_{ij}$
2. Compute $B_i = ||b_i{}^*||^2, i = 1, 2, \ldots, n$
3. for $i = 1, i < n$
    4. If ($\left(\delta - \mu_{i+1,i}{}^2\right)B_i > B_{i+1}$)
        5. Swap $b_i$ and $b_{i+1}$
        6. Go to 1
7. return $b_1, b_2, \ldots, b_n$
```

Example (LLL including GSO)

- LLL ($\delta = \frac{3}{4}$)
- $b_1 = (2,3,14)^T$, $b_2 = (0,7,11)^T$, $b_3 = (0,0,23)^T$.
  - GSO: $b_1{}^* = b_1$, $b_2{}^* = b_2 - \mu_{21} b_1$, $\mu_{21} = \frac{(b_1{}^*, b_2)}{(b_1{}^*, b_1{}^*)} = \frac{21+154}{4+9+196} = \frac{175}{209}$, $\mu_{31} = \frac{322}{209}$, $\mu_{31} = \frac{3473}{4905}$. $b_2{}^* = (-\frac{350}{209}, \frac{938}{209}, -\frac{151}{209})^T$
  - Size reduction: $b_2 = b_2 - \lceil \mu_{21} \rfloor b_1 = (-2,4,-3)^T$, $\mu_{21} = \mu_{21} - \lceil \mu_{21} \rfloor = -\frac{34}{209}$; $b_3 = b_3 - \lceil \mu_{32} \rfloor b_2 = (-2,4,20)^T$, $\mu_{31} = \mu_{31} - \lceil \mu_{31} \rfloor = -\frac{1432}{4905}$; last change is $b_3 = b_3 - \lceil \mu_{31} \rfloor b_1 = (-4,1,6)^T$, $\mu_{31} = \mu_{31} - \lceil \mu_{31} \rfloor = -\frac{79}{209}$.
  - Now, $b_1 = (2,3,14)^T$, $b_2 = (-2,4,-3)^T$, $b_3 = (-4,1,6)^T$.
  - $B_1 = 209$, $B_2 = \frac{4905}{209}$, $B_3 = \frac{103684}{4905}$. Lovasz condition is not satisfied for $i = 1$: since $(\delta - \mu_{21}{}^2) B_1 > B_2$. So swap b_1 and b_2.
  - Applying GSO we get $\mu_{21} = \frac{-34}{29}$, $\mu_{31} = \frac{-6}{29}$, and $\mu_{32} = \frac{2087}{4905}$.
  - Size reduction produces: $b_2 = b_2 - \lceil \mu_{21} \rfloor b_1 = (0,7,11)^T$ and $\mu_{21} = \frac{-6}{29}$. μ_{31} and μ_{32} don't change. μ_{32}

From Heiko Knopse

Example (LLL including GSO) - continued

- Now Lovasz condition is satisfied for $i = 1$ since $(\delta - \ {\mu_{21}}^2)B_1 < B_2$. but not $i = 2$ since $(\delta - \ {\mu_{32}}^2)B_2 < B_3$. swap b_2 and b_3.
  - Now, $b_1 = (-2,4,-3)^T$, $b_2 = (-4,1,6)^T$, $b_3 = (0,7,11)^T$. $B_1 = 29$, $B_2 = \frac{1501}{29}$, $B_3 = \frac{103684}{1501}$. GSO coefficients are $\mu_{21}= \frac{-6}{29}$, $\mu_{31} = \frac{-5}{29}$, and $\mu_{32}= \frac{2087}{1501}$. Applying size reduction does not affect b_2 or μ_{21}. $b_3 = b_3 - \upharpoonright \mu_{32} \downharpoonleft b_2 = (4,6,5\ )^T$, $\mu_{31} = \mu_{31} - \upharpoonright \mu_{32} \downharpoonleft \mu_{21} = \frac{1}{29}$, $\mu_{31}= \frac{586}{1501}$. Both Lovasz conditions now hold.
  - LLL basis is thus $b_1 = (-2,4,-3)^T$, $b_2 = (-4,1,6)^T$, $b_3 = (4,6,5)^T$. Notice $||b_1||$ is actually the shortest vector in Λ.

From Heiko Knopse

LLL Properties

- Suppose we apply LLL to a lattice basis $b_1, b_2, \dots, b_n$ for Λ, ${b_1}^*, {b_2}^*$,…, ${b_n}^*$ and $B_1, B_2, \dots, B_n$ defined as above, we have:
    1. If $X = min_{v\in\Lambda}\left(||b_i||\right)$ and $\frac{1}{4} < \delta$<1, LLL runs in time $O(n^6 \ln(X)^3)$
    2. If $\delta = \frac{3}{4}, B_i \leq 2B_{i+1}$
    3. $B_i \leq \left||b_i|\right|^2 \leq (\frac{1}{2} + 2^{i-2})B_i$
    4. $||b_i|| \leq 2^{\frac{i-1}{2}}||{b_i}^*||$
    5. $\lambda_1(\Lambda) \geq min_i(||{b_i}^*||)$
    6. $||b_1|| \leq 2^{\frac{n-1}{2}}\lambda_1(\Lambda)$
    7. $\det(\Lambda) \leq \prod_{i=1}^{n} ||b_i|| \leq 2^{\frac{n(n-1)}{4}}\det(\Lambda)$
    8. $||b_i|| \leq 2^{\frac{n(n-1)}{4}}\det(\Lambda)^{\frac{1}{n}}$

- If w is a vector in $\mathbb{R}^n$ and the lattice basis for Λ is $b_1, b_2, \dots, b_n$ with $B = [b_1, b_2, \dots, b_n]$, the coefficients for w are $u = B^{-1}(w)$. w is not necessarily in the lattice but if we take each element in u and round it, $B \downharpoonleft B^{-1}(w)$ 1$\in \Lambda$. This is *Babai rounding*

Attack on RSA using LLL

- Attack applies to messages of the form "M xxx" where only "xxx" varies (e.g.- "The key is xxx") and xxx is small.
- From now on, assume $M(x)=B+x$ where B is fixed
  - $|x|<Y$.
  - Not that $E(M(x))=c= (B+x)^3 \pmod n$
  - $f(x)= (B+x)^3-c= x^3+a_2x^2+a_1x+a_0 \pmod n$.
- We want to find x: $f(x)=0 \pmod n$, a solution to this, m, will be the corresponding plaintext.

Attack on RSA using LLL

- To apply LLL, let:
  - v_1= (n, 0, 0, 0),
  - v_2= (0, Yn, 0, 0),
  - v_3= (0, 0, Y^2n, 0),
  - v_4= (a_0 , a_1Y, a_2Y^2, a_3Y^3)
- When we apply LLL, we get a vector, b_1:
  - $||b_1|| \leqq 2^{(3/4)} |\det(v_1, v_2, v_3, v_4)| = 2^{(3/4)} n^{(3/4)} Y^{(3/2)}$ …. *Equation 1*.
- Let $b_1 = c_1v_1 + \ldots + c_4v_4 = (e_0, Ye_1, Y^2e_2, Y^3e_3)$. Then:
  - $e_0 = c_1n+c_4a_0$
  - $e_1 = c_2n+c_4a_1$
  - $e_2= c_3n+c_4a_2$
  - $e_3 = c_4$

Attack on RSA using LLL

- Now set $g(x)= e_3x^3+e_2x^2+e_1x+e_0$.
- From the definition of the e_i, c_4 $f(x)= g(x)$ (mod n), so if m is a solution of f(x) (mod n), $g(m)= c_4$ $f(m)= 0$ (mod n).
- The trick is to regard g as being defined over the real numbers, then the solution can be calculated using an iterative solver.
- If $Y<2^{(7/6)}n^{(1/6)}$, $|g(x)| \leqq 2||b_1||$.
- So, using the Cauchy-Schwartz inequality, $||b_1||\leqq 2^{-1}n$.
- Thus $|g(x)|<n$ and $g(x)=0$ yielding 3 candidates for x.
- Coppersmith extended this to small solutions of polynomials of degree d using a d+1 dimensional lattice by examining the monic polynomial $f(T)= 0$ (mod n) of degree d when $|x|\leqq n^{1/d}$.

Example attack on RSA using LLL

- p= 757285757575769, q= 2545724696579693.
- n= 1927841055428697487157594258917.
- B= 200805000114192305180009190000.
- c= $(B+m)^3$, 0<=m<100.
- f(x)= $(B+x)^3$-c= $x^3+a_2x^2+a_1x+a_0$ (mod n).
  - a_2= 602415000342576915540027570000
  - a_1= 1123549124004247469362171467964
  - a_0= 587324114445679876954457927616
  - v_1= (n,0,0,0)
  - v_2= (0,100n,0,0)
  - v_3= $(0,0,10^4 n,0)$
  - v_4= $(a_0, a_1 100, a_2 10^4, 10^6)$

Example attack on RSA using LLL

- Apply LLL, b_1=
  - 308331465484476402v_1 + 589837092377839611v_2 +
  - 316253828707108264v_3 + (-1012071602751202635)v_4 =
  - (246073430665887186108474, -577816087453534232385300,
    4058485655851944008800000, -1012071602751202635000000)
- g(x)= (-1012071602751202635) t^3 + 40584856558519440088 t^2 +
  (-5778160874535344232385 3) t +246073430665887186108474.

- Roots of g(x) are 42.0000000, (-.9496±76.0796i)
- The answer is 42.

GGH Public Key System

- Pick $n, M \in \mathbb{N}$ and σ is “small”, say $\sigma = 4$
- Plaintext: $\mathcal{M} = \{x: -M \ \leq x \leq M\}$
- Cipherspace: $\mathcal{C} \in \ \mathbb{Z}^n$
- Lattice based but not used.

- Gen:
  1. Choose $B \in \mathbb{Z}^{n \times n}$ with small entries $|B_{ij}| \leq \sigma$
  2. Check B is invertible. B is the secret key.
  3. $H = HNF(B)$
- Enc
  1. For $\vec{m} \in \mathcal{M}$, choose $\vec{r} \in (-\sigma, \sigma)$ uniformly at random
  2. $\vec{c} = H\vec{m} + \vec{r}$
- Dec
  1. Babai round $\vec{m} = H^{-1}Bround((B^{-1}(\vec{c})))$

GGH Example

- $B = \begin{bmatrix} 2 & -3 & 1 & -4 \\ -2 & 1 & 0 & 4 \\ -1 & 3 & 2 & 1 \\ -1 & -4 & 3 & -2 \end{bmatrix}$, H= $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 44 & 18 & 4 & 49 \end{bmatrix}$, $r = (-1,1,1,-1)^T$
- $m = (3,-4,1,3)^T$, $c = Hm + r = (2,-3,2,210)^T$
- $B^{-1} = \frac{1}{49}\begin{bmatrix} 61 & 45 & 10 & -27 \\ -10 & -13 & 8 & -2 \\ 29 & 23 & 16 & -4 \\ 33 & 38 & 3 & -13 \end{bmatrix}$, $B^{-1}c = (-809,-55,-117,-396)^T$
- $H^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \frac{-44}{49} & \frac{-18}{49} & \frac{-4}{49} & \frac{1}{49} \end{bmatrix}$, $m = H^{-1}B \downharpoonleft B^{-1}c \upharpoonleft = (3,-4,1,3)^T$

From Heiko Knopse

LWE

- Based on solving noisy linear equations $mod\ q$. Choose $\overrightarrow{a_i} \in \mathbb{Z}_q{}^n$ uniformly at random. $\vec{s} \in \mathbb{Z}_q{}^n$ is a secret and $m \geq n$ approximate equations $\overrightarrow{a_i} \cdot \vec{s} = b_i\ (mod\ q)$. Errors, $e_1, e_2, \ldots, e_n$ are chosen from distribution χ. Reduces to LWE. Chris Peikert et. al.
- Search LWE problem: Given the above, find $\vec{s}$.
- Decision LWE: Distinguish with non-negligible probability, between $\vec{b} = A\vec{s}$+ $\vec{e}$ and $\vec{b} \in \mathbb{Z}_q{}^m$ chosen uniformly at random given $A, \vec{b}$
- Peikert's results show it is possible to pick parameters so that solving the cipher is equivalent to solving worst-case LWE

LWE cryptosystem

- Given $(n \geq m, l, t, r, q, \chi)$ where χ is a probability distribution $\mathbb{Z}_q$, message space is ${\mathbb{Z}_2}^l$ and cipher space is ${\mathbb{Z}_q}^n \times {\mathbb{Z}_q}^l$.
- Key Gen
  1. Choose $\mathrm{S} \in {\mathbb{Z}_q}^{n \times l}$, uniformly from the distribution χ.
  2. Choose $A \in {\mathbb{Z}_q}^{m \times n}$, and $\mathrm{E} \in {\mathbb{Z}_q}^{m \times l}$ uniformly from the distribution χ.
  3. Private key is S, public key is $(A, P = AS + E)$
- Enc
  1. For $\vec{v} \in {\mathbb{Z}_2}^l$, choose $\vec{a} \in \{0,1\}^m$, uniformly at random
  2. $\overrightarrow{CT} = (\vec{u} = A^T\vec{a}, \vec{c} = P^T\vec{a} + \upharpoonright \frac{q}{2} \downharpoonleft \vec{v}$))
- Dec
  1. Compute $\upharpoonright (\upharpoonright \frac{q}{2} \downharpoonleft)^{-1}(\vec{c} - S^T\vec{u}) \upharpoonleft (mod\ 2)$

- Decryption may have errors. Suppose χ is a discrete Gaussian $D_{\mathbb{Z},S}$. Then $E^T\vec{a}$ has magnitude $\leq \sqrt{m}s$ with high probability. Error occurs if $E^T\vec{a} \geq \frac{q}{4}$. One can show that for any n, $\exists q, m, s$ such that the error is small and the underlying LWE problem is hard.

LWE example

- $n = 4, q = 23, m = 8, \alpha = \frac{5}{23}, s = 5, \sigma = \frac{s}{\sqrt{2\pi}}$

- $A = \begin{bmatrix} 9 & 5 & 11 & 13 \\ 13 & 6 & 6 & 2 \\ 6 & 21 & 17 & 18 \\ 22 & 19 & 20 & 8 \\ 2 & 17 & 10 & 21 \\ 10 & 8 & 17 & 11 \\ 5 & 16 & 12 & 2 \\ 5 & 7 & 11 & 7 \end{bmatrix}, S = \begin{bmatrix} 5 & 2 & 9 & 1 \\ 6 & 8 & 19 & 1 \\ 19 & 18 & 9 & 18 \\ 9 & 2 & 14 & 18 \end{bmatrix}$

From Heiko Knopse

LWE example

- $E = \begin{bmatrix} 0 & 22 & 1 & 21 \\ 0 & 22 & 22 & 22 \\ 6 & 21 & 17 & 18 \\ 22 & 22 & 22 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 2 \\ 1 & 22 & 1 & 22 \\ 22 & 0 & 0 & 1 \end{bmatrix}, \mathrm{P} = \begin{bmatrix} 10 & 5 & 21 & 7 \\ 3 & 1 & 13 & 1 \\ 19 & 15 & 6 & 13 \\ 22 & 22 & 22 & 0 \\ 9 & 20 & 20 & 17 \\ 15 & 21 & 1 & 2 \\ 0 & 12 & 3 & 19 \\ 16 & 2 & 7 & 15 \end{bmatrix},$

From Heiko Knopse

LWE example

- Encrypt $m = (1,0,1,1)^T$, using a $= (1,1,0,1,0,0,0,1)^T$
  - $\lfloor \frac{23}{2} m \rceil = (12,0,12,12)^T$,
  - $(u, c) = \left(A^T a, P^T a + \lfloor \frac{23}{2} m \rceil\right) = ((3,14,2,7)^T, (4,5,7,5)^T)(mod\ 23)$

- Decrypt:
  - $m' = c - S^T u = (11,21,12,10)^T\ (mod\ 23)$,
  - $\lfloor \frac{1}{12} m' \rceil\ (mod\ 2) = (1,0,1,1)^T$

From Heiko Knopse

LWE example

- Encrypt $m = (1,0,1,1)^T$, using a $= (1,1,0,1,0,0,0,1)^T$
  - $\downharpoonright \frac{23}{2} m \upharpoonleft = \ (12,0,12,12)^T$,
  - $(u, c) = \left(A^T a, P^T a + \downharpoonright \frac{23}{2} m \upharpoonleft\right) =$
    $((3,14,2,7)^T, (4,5,7,5)^T)(mod\ 23)$

- Decrypt:
  - $m' = c - S^T u = (11,21,12,10)^T\ (mod\ 23)$,
  - $\downharpoonright \frac{1}{12} m' \upharpoonleft\ (mod\ 2) = (1,0,1,1)^T$

From Heiko Knopse

LWE/Ring-LWE parameters

Level	n	q	s	P	P&A	c	Exp
Low	128	4093	8.87	2.9×10^5	7.4×10^5	3.8×10^3	30
High	320	4093	8	4.9×10^5	17.7×10^5	17.4×10^3	42

- Ring-LWE cuts ciphertext by factor of n

Ring-LWE

- Put $R = R_q = \frac{\mathbb{Z}_q[x]}{x^n+1}, n = 2^k, R \approx \mathbb{Z}_q{}^n.$ $a \in R$, generates ideal (a) corresponding to a q-ary ideal lattice.

- Ring LWE: Given a $\in R$, and $b = as + e$, for s, e $\in R$, find s.

- Solving R-LWE is at least as hard as solving CVP_γ on arbitrary ideal lattices

NTRU Public Key System

- NTRU is a ring lattice-based system.
- $R = \frac{\mathbb{Z}[x]}{x^N - 1}, R_p = \frac{\mathbb{Z}_p[x]}{x^N - 1}, R_q = \frac{\mathbb{Z}_q[x]}{x^N - 1}$
- $(c_0 + c_1 x + \cdots + c_{N-1}) = (a_0 + a_1 x + \cdots + a_{N-1}) \otimes (b_0 + b_1 x + \cdots + b_{N-1})$, where $c_k = \sum_{i+j=k\ (mod\ N)} a_i b_j$
- $\mathcal{T}(d_1, d_2)$ is the set of "ternary" polynomials of degree $< N$, having d_1 coefficients equal to 1, having d_2 coefficients equal to -1, and remaining coefficients equal to 0.
- Pick N, p prime and $q, d \in \mathbb{N}$, $(p, q) = (N, q) = 1$, $q > (6d + 1)p$.

NTRU Public Key System

- KeyGen
  1. Pick $f, g \in R, f \in \mathcal{T}(d+1, d), g \in \mathcal{T}(d, d)$.
  2. Find f_p, f_q: $f \cdot f_p = 1\ (mod\ p),\ \ f \cdot f_q = 1\ (mod\ q), \mathrm{h} = f_q \cdot g\ (mod\ q)$.
  3. Public key is (N, p, q, h), private key is f.
- Plaintext is $m \in R_p$, ciphertext is $c \in R_q$
- Encryption
  1. Chose random $r \in R, r \in \mathcal{T}(d, d)$.
  2. $c = prh + m\ (mod\ q)$.
- Decryption
  1. Compute $a = fc\ (mod\ q)$
  2. Plaintext is f_pa.
  3. Verify that $a = fc = f(prh + m)(mod\ q) = pfrf_q g + fm(mod\ q) = prg + fm(mod\ q)$.

NTRU Example

- $N = 5, p = 3, q = 29, d = 1, f = x^4 + x^3 - 1, g = x^3 - x^2$
- $f_p = -x^3 - x^2 + x - 1, f_q = -5x^4 + 8x^3 + 3x^2 + 11x + 13$
- $h = f_q g = 8x^4 + 2x^3 + 11x^2 + 13x - 5\ (mod\ 29)$
- $r = x^4 - x$
- $c = prh + m = 8x^4 + 21x^3 + 25x^2 + 20x + 15\ (mod\ 29)$
- $a = fc = -2x^4 + 2x^3 + 4x^2 - 3x + 1\ (mod\ 29)$
- We check $a = prg + fm$ in R
- $m = x^3 + x$

From Heiko Knopse

Some NIST Round 3 entries

- Public-Key Encryption/KEMs
  - Classic McEliece
  - CRYSTALS-KYBER
  - NTRU
  - SABER
- Digital Signatures
  - CRYSTALS-DILITHIUM
  - FALCON
  - Rainbow

- Public-Key Encryption/KEMs (Alternates)
  - BIKE;
  - FrodoKEM
  - HQC
  - NTRU Prime
  - SIKE
- Digital Signatures
  - GeMSS
  - Picnic
  - SPHINCS+

Winner: Dilithium (signing), Kyber (key-encapsulation)

Some common features of Dilithium and Kyber

- Ring is $\mathbb{Z}_p[x]/(x^{256}+1)$ in both cases
  - $p = 3329$ for Kyber
  - $p = 2^{23} - 2^{13} + 1$ for Dilithium
  - So, the same modular arithmetic we all grew up with.
- For both primes, there is a primitive (and hence 128 primitive) 256th roots of unity (You are not expected to understand this).
  - As a result, $x^{256} + 1$ factors into coprime 128 quadratics
  - Allows us to perform a "Number Theory Transform" that turns convolution into pointwise multiplication for ring operations giving a nice speedup

Useful definitions

- $r^+ = r \ (mod \ q), q > r^+ \geq 0$
- $r' = r \ mod^{\pm}(m)$ means $r' = r \ (mod \ m)$ and $-\frac{m}{2} \leq r' \leq \frac{m}{2}$

- $decompose(r, \alpha, q)$
  - $r_0 = \ r^+ \ mod^{\pm}(\alpha)$
  - if $r^+ - r_0 == (q-1)$
    - $r_1 = 0 \ , r_0 = q - 1$
  - else
    - $r_1 = \frac{r^+ - r_0}{\alpha}$
  - return (r_1, r_0)

Useful definitions

- $lowbits(x, \alpha, q)$
  - $(r_1, r_0) = decompose(x, \alpha, q)$
  - return r_0

- $highbits(x, \alpha, q)$
  - $(r_1, r_0) = decompose(x, \alpha, q)$
  - return r_1

- $power2round(r, d, q)$
  - $r^+ = r\ mod(q)$
  - $r_0 = \ r^+\ mod^{\pm}(2^d)$
  - return $(\frac{r^+ - r_0}{2^d}, r_0)$

Useful definitions

```
• $encode_d(x)$, $x$ is an array of length 256, $m = 2^d$, $1 \leq d \leq 12$
        – for (i = 0; i < 256; i++)
            • $a = x[i]$
            • for (j=0; j < d; j++)
              – $b[d \cdot i + j] = a \ (mod\ 2)$
              – $a = \frac{a - b[d \cdot i + j]}{2}$
    • return bits-to-bytes(b)

• $decode_d(x)$, $x$ is a byte array of length $32d$, $m = 2^d$, $1 \leq d \leq 12$
    • $b = bytes - to - bits(x)$
    • for (i=0; i < 256; i++)
        – $out[i] = \sum_{j=0}^{d-1} b[i \cdot d + j] \cdot 2^j$
    • return $out$
```

Examples

- $decompose(r, \alpha, q)$ examples (second shows roundoff edge case)

$\boldsymbol{q}$	$\boldsymbol{\alpha}$	$\boldsymbol{r}$	$\boldsymbol{r\ mod^{\pm}(\alpha)}$	$\mathbf{r} - \boldsymbol{r\ mod^{\pm}(\alpha)}$	$\boldsymbol{r_0}$	$\boldsymbol{r_1}$
17	8	5	-3	8	-3	1
17	8	15	-1	16	-2	0
3329	104	50	50	0	50	0
3329	104	100	-4	104	-4	1

SHAKE-256/SHAKE-128

- $H(v, d) = SHAKE256(v, d)$
- $H_{128}(v, d) = SHAKE128(v, d)$

- $RAWSHAKE256(J, d) = KECCAK[512](J|11, d)$
- $SHAKE256(M, d) = RAWSHAKE256(M|11, d)$
- $RAWSHAKE128(J, d) = KECCAK[256](J|11, d)$
- $SHAKE128(M, d) = RAWSHAKE128(M|11, d)$

- Note
  - $SHA3_{256}(M) = KECCAK[512](M|11{,}256)$
  - $SHA3_{512}(M) = KECCAK[1024](M|11{,}1024)$

Number Theory Transform (NTT)

- For $p = 3329$, $\mathbb{Z}_p$ has a primitive $256th$ root of unity ($\zeta = 17$), and no 512 root of unity, so $x^{256} + 1$ factors into 128 coprime quadratic factors of the form $(x^2 - \xi)$, $17^{128} = -1$.
- In fact, $x^{256} + 1 = \prod_{k=0}^{127}(x^2 - \zeta^{2\cdot bitrev_7(k)+1})$. $bitrev_7(k)$ simply reverses the bit order in a 7-bit byte, k.
  - $x^{256} + 1 = (x^2 - 17) \cdot (x^2 - 17^{129}) \cdot \cdots \cdot (x^2 - 17^{255})$

- For $p = 8380417$, $\zeta = 1753$.

- Because of this, an analog of the Chinese remainder theorem holds in $R_p = {}^{\mathbb{Z}_p[x]}\!\big/_{(x^{256}+1)}$.

Dilithium (simplified)

- Remember $A^{k\times l}$ is generated randomly from $R = \mathbb{Z}_p[x]/(x^{256}+1)$.
- s_1 is a vector of dimension l with entries from R has random coefficients $\leq \eta$
- s_2 is a vector of dimension k with entries from R has random coefficients $\leq \eta$
- $t = As_1 + s_2$

```
Sign
    $y \coloneqq {S_{\gamma_1 - 1}}^l$
    $w_1 \coloneqq \mathrm{highbits}(\mathrm{Ay}, 2\gamma_2)$
    $c \coloneqq SH(M||w_1)$
    $z \coloneqq y + cs_1$
    return($z, c$)
```

```
Verify
    ${w_1}' \coloneqq \mathrm{highbits}(\mathrm{Az} - \mathrm{ct}, 2\gamma_2)$
    $c' \coloneqq SH(M||{w_1}')$
    Check $c' == c$ AND $||z||_\infty <$
    $\gamma_1 - \beta$
```

Dilithium (less simplified)

Parameters: $p = 8380417, \;\; k = 5, l = 4, \gamma_1 = \frac{p-1}{16}, \gamma_2 = \frac{\gamma_1}{2}, \eta = 5, \beta = 275$

- KeyGen
  - $A \in R^{k \times l}$, selected from random distribution over R_p
  - $(s_1, s_2) \in {S_\eta}^k \times {S_\eta}^l$, selected at random, ${S_\eta}^k$ consists of elements of R^k with coefficients $\leq \eta$
  - Set $t = As_1 + s_2$
  - Public key is (A, t), Private key is (s_1, s_2)

For the sake of compression A is generated from a seed and SHAKE-256

Dilithium

- Sign(pk, sk, M) --- simplified

```
$z = \perp$
while ($z = \perp$) {
    $y = {S_{\gamma_1}}^{l} - 1$
    $w_1 = highbits(Ay, 2\gamma_2)$
    $c = SHAKE - 256(M||w_1)$
    $z = y + cs_2$
    if ($||z||_\infty \geq \gamma_1 - \beta$) OR $lowbits(Ay - cs_2, 2\gamma_1) \geq \gamma_2 - \beta$) then $z = \perp$
}
```

Signature is (z, c)

- Real Dilithium uses a number of functions to generate A from a seed. It also has a hedged version and a deterministic version. The hedged version avoids some possible side channels.

Dilithium

- Verify(pk, M,z, c) --- simplified
  1. $w_1' = highbits(Az - ct, 2\gamma_2)$
  2. Return true if $||z||_\infty \leq \gamma_1 - \beta$ AND $c = SHAKE - 256(M||w_1')$, otherwise return false

Kyber

- Parameters: $\left(p = 3329, R = \frac{\mathbb{Z}_p}{x^{256}+1}, k = 4, \eta = 2\right)$, $\hat{x} = NTT(x)$

- KeyGen
  1. $\hat{t} = \hat{A}\hat{s} + \hat{e}$, A is generated from seed ρ
- Encrypt(m, r) [$r \in R^k$ is generated from CDB_{η_1} , e_1 is generated from CDB_{η_2}
  1. $u(x) = NTT^{-1}\left(\hat{A}^T\right) + e_1$
  2. $\mu = decompress_1(decode_1(m))$, $v = NTT^{-1}(\hat{t}^T \cdot r + e_1 + \mu)$
  3. $c_1 = encode_{d_u}(compress_{d_u}(u))$, $c_2 = encode_{d_v}\left(compress_{d_v}(r)\right)$
  4. Return (c_1, c_2)
- Decrypt(c_1, c_2)
  1. $w = v - NTT^{-1}(\hat{s} \cdot NTT(u))$
  2. Return $encode_1(compress_1(w))$

Kyber

- Parameters: $(p = 3329, R = \frac{\mathbb{Z}_p}{x^{256}+1}, k = 4, \eta = 2), \hat{x} = NTT(x)$

- EKM-KeyGen
  1. z is a random 32-byte value
  2. $(e_{PKE}, d_{PKE}) = KeyGen$
  3. $d_{KEM} = d_{PKE}||H(e_{PKE})||z$
  4. Return (e_{PKE}, d_{KEM})

Note $H(x) = SHA3_{256}(x)$

Kyber

- Parameters: $(p = 3329, R = \frac{\mathbb{Z}_p}{x^{256}+1}, k = 4, \eta = 2), \hat{x} = NTT(x)$
- EKM-Encaps
  1. m is a random 32-byte value
  2. $(K, r) = SHA - 3_{512}(m||H(e_h))$
  3. $c = kyber - encrypt(ek, m, r)$
  4. Return (K, c)
- EKM-Decaps
  1. $m' = kyber - decrypt(dk, c)$
  2. $(K', r') = SHA - 3_{512}(m'||H(e_k))$
  3. $\overline{K} = SHAKE - 256(z||c, 32)$
  4. $c' = kyber - encrypt(e_k, m', r')$
  5. If ($c == c'$) return K′ else error

Useful definitions

- $Usehint, makehint$
- $Smallball, etc$
- $SamplePolyCBD_{\eta}(x)$
- $PRF_{\eta}(x)$
- $SampleNTT(x)$
- MultiplyNTT
- BaseCaseMultiply
- NTT
- NTT^{-1}

Dilithium unedited

- x

Kyber unedited

- x

End

Endomorphisms

- Endomorphisms are homomorphisms from E(K) → E(K) that can be represented by rational functions.
  - If $a(x,y)=(r_1(x), r_2(x)y)$, $r_1(x)= p(x)/q(x)$. deg(a)= max(deg(p), deg(q)).
  - The endomorphism, a, is separable, if r’(x)¹0.
    - If a is separable deg(a)= #ker(a).
    - If a is not separable deg(a)> #ker(a).

- If f_p is the Frobenius map, it is an endomorphism of degree p and f_p is not separable.
  - $\ker(f_p\text{-}1)=\#E_p$. f_p-1 is a separable endomorphism.
  - Let E be an elliptic curve over F_p, a= q+1-$\#E_p$= q+1- deg(ker(f_p-1)). f_p^2-af_p+q=0.

Shanks and Menstre

- Input: $E_q(a,b)$, $\#E_q(a,b)=q+1-t$, $|t| \leqq 4<q$.
- Output: Bound on t. $O(q^{1/4}+e)$.
    1. Pick random point P on $E_q(a,b)$, $|P|>4<q$.
    2. $Q=[q+1]P$
    3. $Q_1= Q+ \text{floor}[2q]P$
    4. $t' = t+ \text{floor}[2q]$, note $0 \leqq t' \leqq 4q$
    5. $m= \text{ceiling}(2q^{1/4})$
    6. Baby step: $[j]P$
    7. Giant step: $Q_1-[i][m]P$
    8. $t' = im+j$, $i,j<m$. This bounds $\#E_q(a,b)$.
- Menstre: either a curve or its twist has a point with order >4q

Endomorphisms continued

- Endomorphism are maps that preserve the “addition” operation between an elliptic curve group and itself. That is j(P+Q)= j(P)+j(Q). We care about endomorphisms that preserve O: j (O)= O. These are called isogonies.

- There are two very important endomophisms:
  - Frobenius: j(x,y)= (x^p, y^p)
  - Point multiplication: j(x,y)= [n](x,y).

- For $E_K(a,b)$, define Δ= (-16)($4a^3+27b^2$). (For singular curves Δ=0) and define the j-invariant $E_p(a,b)$, $j(E) = \frac{1728}{\Delta}$.

Isomorphic Curves and the j-invariant

- Let K be a field and K* its algebraic closure. $E_K(a,b)$ and $E_K(a',b')$ are *isomorphic* if r,s,t **ε** K, u **ε** K*: the transformations (x,y) → (x',y') given by $x=u^2x'+r$, $y= u^3y'+su^2x'+t$, take $E_K(a,b)$ to $E_K(a',b')$.

- Recall $D= (-16)(4a^3+27b^2)$. (For singular curves $\Delta=0$) and define the j-invariant $E_p(a,b)$, $j(E)= 1728/\Delta$.

- Theorem: Let $E_1=E_K(a,b)$ and $E_2=E_K(a',b')$ be two ellliptic curves.
  1. If E_1 and E_2 are isomorphic, they have the same j-invariant.
  2. If $j(E_1)=j(E_2)$, there is a m: $a_2= \mu^4 a_1$, $b_2= \mu^6 b_1$.
  3. If two curves have the same j-invariant, they are isomorphic over the algebraic closure, K*.

The Division Polynomials

- [m] $(x,y)= (q_m(x,y)/y_m(x,y)^2, w_m(x,y)/y(x,y)^3)$
- We can calculate these polynomials recursively:
  - $y_0(x,y)= 0$; $y_1(x,y)= 0$.
  - then $y_{2m+1}(x,y)= y_{m+2}(x,y)y_m{}^3+y_{m-1}(x,y)y_{m+1}{}^3$.
  - $f_m=xy_m{}^2-y_{m+1}y_{m-1}$
  - $w_m= 1/(4y)(y_{m+2}y_{m-1}{}^2–y_{m-2}y_{m+1}{}^2)$

- Let E be an elliptic curve, the endomorphism of E given by multiplication by n has degree n^2.

- (x,y)=P **ε** E[m] is the subgroup of torsion points whose order divides m: [m]P=0.

Preliminary DSA

- Bob has a private key x and a public key <g,X>: $X= g^x$ in a group G. To sign m, given a map f: G → $Z_{|G|}$:
  1. Bob generates a random a: $1 \leqq a < |G|$. $A= g^a$.
  2. Bob computes $B \varepsilon Z_{|G|}$: $m= -xf(A)+Ba \pmod{|G|}$.
  3. $Sig_{Bob}(m)= (A,B)$

- To verify compute $u= mB^{-1} \pmod{|G|}$, $v=f(A)B^{-1} \pmod{|G|}$ and $w=g^uX^v$. Verify that w=A.

ECDSA

- D=(q, a,b,P,n,h). nh= #E_q(a,b). Private key d, message m.

- Signature (r,s)
  1. Select k**ε**[1,n-1]
  2. Compute kP=(x_1, y_1). Convert x_1 to integer x_1.
  3. Compute r= x_1 (mod n). If r=0 goto 1.
  4. Compute e=H(m).
  5. s= k^{-1}(e+dr) (mod n). If s=0, goto 1.

- Verify
  1. Check r,s [1,n-1]. Compute e=H(m).
  2. Compute w= s^{-1} (mod n). u_1= ew (mod n). u_2= rw (mod n).
  3. Compute X= u_1 P+ u_2 Q. If X= O, reject.
  4. Convert x_1 of X to integer x_1. Compute v= x_1 (mod n).
  5. If (v=r) accept signature.

ECIES

- Input D=(q, a, b, P, n, h), public key Q, plaintext m.
- ENC, MAC, DEC are standard "symmetric key" functions. KDF is key derivation function (also standard).

1. Pick k**ε**[1, n-1].
2. Compute R= kP, Z=hkQ. If Z=O, go to 1.
3. (k[1], k[2]) = KDF(x_Z, R).
4. c= ENC_{k1}(m), t= $\text{MAC}_{k[2]}$(c).
5. return (R, c, t)

LLL

- Definition: $B= \{b_1, \ldots, b_n\}$, L in R^n. $m_{i,j}= (b_i, b_j^*)/(b_j^*, b_j^*)$. $b_i^*= b_i- \sum_{j=1}^{i-1} m_{i,j} b_j^*$. B is *reduced* if
  1. $| m_{i,j} | \leqq 1/2$; $1 \leqq j < i \leqq n$
  2. $||b_i^*||^2 (3/4-m_{i,i-1}^2)||b_{i-1}^*||^2$.

- Note $b_1^*=b_1$.

LLL algorithm

```
$b_1$*= $b_1$; k= 2;
for(i=2; i≦n; i++) {
    $b_i$*= $b_i$;
    for(j=1; j<i; j++)
    {   $□_{i,j}$= ($b_i$ , $b_j$*)/$B_j$;
        $b_i$*= $b_i$−□$□_{i,j}b_j$*;$B_i$= ($b_i$*, $b_i$*);}
}
for(;;) {
    RED(k, k-1);
    if($B_k$<(3/4 – $□_{k,k-1}^2$)$B_{k-1}$) {
        □= $□_{k,k-1}$; B= $B_k$+□□$□^□B_{k-1}$;$□_{k,k-1}$= □□$B_{k-1}$/B;
        $B_k$= $B_{k-1}B_k$/B; $B_{k-1}$= B; swap($b_k$, $b_{k-1}$);
        if(k>2) swap($b_k$, $b_{k-1}$);
        for(i=k+1; i□□n;i++)
        {   t= $□_{i,k}$;; $□_{i,k}$;= $□_{i,k-1}$−□□t;
            $□_{i,k-1}$=t+□$□_{k,k-1}$□$□_{i,k}$; }
        k= max(2, k-1);
        if(k>n)  return($b_1$, …, $b_n$);
}
```

```
RED(k, k-1)

if(|$□_{k,l}$|)> 1/2) {
    r= □1/2+$□_{k,l}$□;
    $b_k$= $b_k$ −r $b_l$;
    for(j=1; j<l;j++) {
        $□_{k,j}$= $□_{k,j}$−r$□_{l,j}$;
        $□_{k,l}$ = $□_{k,l}$−r;
    }
}
```

LLL Theorem

- Let L be the n-dimensional lattice generated by <v_1, …, v_n> and l the length of the shortest vector in L. The LLL algorithm produces a reduced basis <b_1, …, b_n> of L.

  1. $||b_1|| \leqq 2^{(n-1)/4} D^{1/n}$.
  2. $||b_1|| \leqq 2^{(n-1)/2} l$.
  3. $||b_1||\ ||b_2||\ \dots\ ||b_n|| \leqq 2^{n(n-1)/4} D$.

- If $||b_i||^2 \leq C$ algorithm takes $O(n^4 \lg(C))$.

Gauss again

- Let <v_1, v_2> be a basis for a two dimensional lattice L in R^2. The following algorithm produces a reduced basis.

```
for(;;) {
    if(||$v_1$||>||$v_2$||)
        swap $v_1$ and $v_2$;
    t= [($v_1$, $v_2$)/($v_1$, $v_1$)]; // [] is the “closest integer”
      function
    if(t==0)
        return;
    $v_2$ = $v_2$-t$v_1$;
    }
```

- <v_1, v_2> is now a reduced basis and v_1 is a shortest vector in the lattice.