Lecture 14: Elliptic Curve Cryptography and Digital Rights Management

Lecture Notes on "Computer and Network Security"

by Avi Kak (kak@purdue.edu)

April 20, 2011

12:50pm

Goals:

- Introduction to elliptic curves
- A group structure imposed on the points on an elliptic curve
- Geometric and algebraic interpretations of the group operator
- Elliptic curves on prime finite fields
- Elliptic curves on Galois fields
- Elliptic curve cryptography
- Security of Elliptic Curve Cryptography
- ECC for Digital Rights Management (DRM)

14.1: Why Elliptic Curve Cryptography?

- As you saw in Section 12.8 of Lecture 12, the computational overhead of the RSA-based approach to public-key cryptography increases with the size of the keys. As algorithms for integer factorization have become more and more efficient, the RSA based methods have had to resort to longer and longer keys.

- Elliptic curve cryptography (ECC) can provide the same level and type of security as RSA (or Diffie-Hellman as used in the manner described in Section 13.5 of Lecture 13) **but with much shorter keys**.

- Table 1 compares the key sizes for three different approaches to encryption for comparable levels of security against brute-force attacks. What makes this table all the more significant is that for comparable key lengths the *computational burdens* of RSA and ECC are comparable. *What that implies is that, with ECC, it takes one-sixth the computational effort to provide the same level of cryptographic security that you get with 1024-bit RSA.* [The table shown here is basically the same table as presented earlier in Section 12.9 of Lecture 12, except that now we also include ECC in our comparison.]

Symmetric Encryption *Key Size* *in bits*	*RSA and Diffie-Hellman* *"Key" size* *in bits*	*ECC* *"Key" Size* *in bits*
80	1024	160
112	2048	224
128	3072	256
192	7680	384
256	15360	512

Table 1: *A comparison of key sizes needed to achieve equivalent level of security with three different methods.*

- The computational overhead of both RSA and ECC grows as $O(N^3)$ where N is the key length in bits. [Source: Hank van Tilborg, NAW, 2001] Nonetheless, despite this parity in the dependence of the computational effort on key size, it takes far less computational overhead to use ECC on account of the fact that you can get away with much shorter keys.

- Because of the much smaller key sizes involved, ECC algorithms can be implemented on **smartcards** without mathematical co-processors. **Contactless smart cards** work only with ECC because other systems require too much induction energy. Since shorter key lengths translate into faster handshaking protocols, ECC is also becoming increasingly important for **wireless communications**. [Source: Hank van Tilborg, NAW, 2001]

- For the same reasons as listed above, we can also expect ECC to become important for **wireless sensor networks**.

- ECC is also used in the algorithms for Digital Rights Management (DRM), as we will discuss in Section 14.14.

- In case the reader is wondering why we placed the word *key* between quotation marks in the header of the RSA column in Table 1, read the beginning of Section 12.8 of Lecture 12. The reason for quoting the same word in the header for the ECC column will turn out to be similar, as you will see from this lecture.

14.2: The Main Idea of ECC — In a Nutshell

- Imagine we have a set of points (x_i, y_i) in a plane. The set is very, very large but finite. We will denote this set by E.

- Next imagine we can define a group operator on this set. As you know from Lecture 4, a group operator is typically denoted by the symbol '+' even when the operation itself has nothing whatsoever to do with ordinary arithmetic addition. So given two points P and Q in the set E, the group operator will allow us to calculate a third point R, also in the set E, such that $P + Q = R$.

- Given a point $P \in E$, we will particularly be interested in using the group operator to find $P+P$, $P+P+P$, $P+P+P+\ldots+P$ for an arbitrary number of repeated invocations of the group operator. Given *an ordinary integer* k, we will use the notation $k \times P$ to express adding P to itself k times. [Note that $k \times P$ is NOT an attempt to define a multiplication operator on the set E. That is because k is an ordinary integer. In other words, k is not in the set E. The only meaning to be associated with $k \times P$ is that of repeated addition.]

- Now imagine that the set E is magical in the sense that, after we have calculated $k \times P$ for a given point $P \in E$, it is extremely difficult to recover k from $k \times P$. We will assume that the only way to recover k from $k \times P$ is to try every possible repeated summation like $P + P$, $P + P + P$, $P + P + P + \ldots + P$ until the result equals what we have for $k \times P$. [Trying to figure out how many times an element P must be added to itself in $P + P + P + \ldots + P$ in order for the result to equal $k \times P$ is referred to as solving the *discrete logarithm problem*. To see why that is so, consider the traditional notion of logarithm that allows us to write $a^k = b$ as $k = \log_a b$. But, obviously, a^k is nothing but $a \times a \times \ldots \times a$ with the application of the '×'operator repeated k times. So when we write $a^k = b$ as $k = \log_a b$, we calculate the number of times the operator '×' was repeated on the element a. That is the same thing as what we want to do in order to determine the value of k from $k \times P$. Just don't be fooled by the appearance of the operator '×' in $k \times P$. It is really not a multiplication. It is a shortcut for denoting k repeated additions of P to itself. The notion of discrete logarithms was discussed earlier in Section 11.8 of Lecture 11.]

- If we could ensure the above condition, then "products" like $k \times P$ for $P \in E$ could be used by two parties in a Diffie-Hellman like protocol for sharing a secret session key. Section 14.11 will show you how that can be done.

- **All of the assumptions we have made above are satisfied when the set E of points (x_i, y_i) is drawn from an elliptic curve.**

14.3: What are Elliptic Curves?

- First and foremost, elliptic curves have nothing to do with ellipses. Ellipses are formed by quadratic curves. Elliptic curves are always cubic. [Note: Elliptic curves are called **elliptic** because of their relationship to **elliptic integrals** in mathematics. An elliptic integral can be used to determine the arc length of an ellipse.]

- The simplest possible "curves" are, of course, straight lines.

- The next simplest possible curves are conics, these being quadratic forms of the following sort

$$ax^2 \quad + \quad bxy \quad + \quad cy^2 \quad + \quad dx \quad + \quad ey \quad + \quad f \quad = \quad 0$$

  If $b^2 - 4ac$ is less than 0, then the curve is either an ellipse, or a circle, or a point, or the curve does not exist; if it is equal to 0, then we have either a parabola, or two parallel lines, or no curve at all; if it is greater than 0, then we either have a hyperbola or two intersecting lines. (Note that, by definition, a conic is the intersection of a plane and a cone.)

- The next simplest possible curves are elliptic curves. An elliptic curve in its "standard form" is described by

  $$y^2 \quad = \quad x^3 \; + \; ax \; + \; b$$

  for some fixed values for the parameters a and b. This equation is also referred to as **Weierstrass Equation** of **characteristic** 0. [The equation shown involves multiplications and additions over certain objects that are represented by x, y, a, and b. The values that these object acquire are meant to be drawn from a set that must at least be a **ring** with a multiplicative identity element. (See Lecture 4 for what a ring is.) The **characteristic** of such a ring is the number of times you must add the multiplicative identity element in order to get the additive identity element. If adding the multiplicative identity element to itself, no matter how many times, **never** gives us the additive identity element, we say the characteristic is 0. For illustration, the set of all real numbers is of characteristic 0 because no matter how many times you add 1 to itself, you will never get a 0. When a set is `not` of characteristic 0, there will exist an integer p such that $p \times n = 0$ for all n. The value of p is then the characteristic of the integral domain. For example, in the set of remainders Z_9 (which is a ring with a multiplicative identity element of 1, although it is not an integral domain since $3 \times 3 = 0 \; mod \; 9$) that you saw in Lecture 5, the numbers $9 \times n$ are 0 for every value of the integer n. So we can say that Z_9 is a ring of characteristic 9. When we say that the equation shown above is of characteristic 0, we mean that the set of numbers that satisfy the equation constitutes a ring of characteristic 0.] Elliptic curves have a rich algebraic structure that can be put to use for cryptography.

- Figure 1 shows some elliptic curves for a set of parameters $(a, \; b)$.

The top four curves all look smooth (they do not have cusps, for example) because they all satisfy the following condition on the **discriminant** of the polynomial $f(x) = x^3 + ax + b$:

$$4a^3 + 27b^2 \neq 0 \qquad (1)$$

[Note: The **discriminant of a polynomial** is the product of the squares of the differences of the polynomial roots. The roots of the polynomial $f(x) = x^3 + ax + b$ are obtained by solving the equation $x^3 + ax + b = 0$. Since this is a cubic polynomial, it will in general have three roots. Let's call them r_1, r_2, and r_3. Its discriminant will therefore be

$$D_3 = \prod_{i<j}^{3} (r_i - r_j)^2$$

which is the same as $(r_1 - r_2)^2 (r_1 - r_3)^2 (r_2 - r_3)^2$. It can be shown that when the polynomial is $x^3 + ax + b$, the discriminant reduces to

$$D_3 = -16(4a^3 + 27b^2)$$

This discriminant must not become zero for an elliptic curve polynomial $x^3 + ax + b$ to possess three distinct roots. If the discriminant is zero, that would imply that two or more roots have coalesced, giving the curve a cusp or some other form of non-smoothness. Non-smooth curves are called **singular**. This notion will be defined more precisely later. It is **not safe** to use singular curves for cryptography. AS to why that is the case will become clear later in these lecture notes.]

- The **bottom two** examples in Figure 1 show two elliptic curves for which the condition on the discriminant is violated. For the one on the left that corresponds to $f(x) = x^3$, all three roots of the cubic polynomial have coalesced into a single point and we get a cusp at that point. For the one on the right that corresponds to $f(x) = x^3 - 3x + 2$, two of the roots have coalesced into the point where the curve crosses itself. These two curves are **singular**. As mentioned earlier, it is **not safe** to use singular curves for cryptography.

- Note that since we can write

$$y \quad = \quad \pm\sqrt{x^3 \quad + \quad ax \quad + \quad b}$$

elliptic curves in their standard form will be symmetric about the x-axis.

- It is difficult to comprehend the structure of the curves that involve polynomials of degree greater than 3.

- To give the reader a taste of the parameters used in elliptic curves meant for real security, here is an example:

$$\begin{aligned} y^2 \quad = \quad x^3 \quad & + \quad 317689081251325503476317476413827693272746955927x \\ & + \quad 79052896607878758718120572025718535432100651934 \end{aligned}$$

This elliptic curve is used in the Microsoft Windows Media **Digital Rights Management** Version 2. We will have more to say about this curve in Section 14.13.

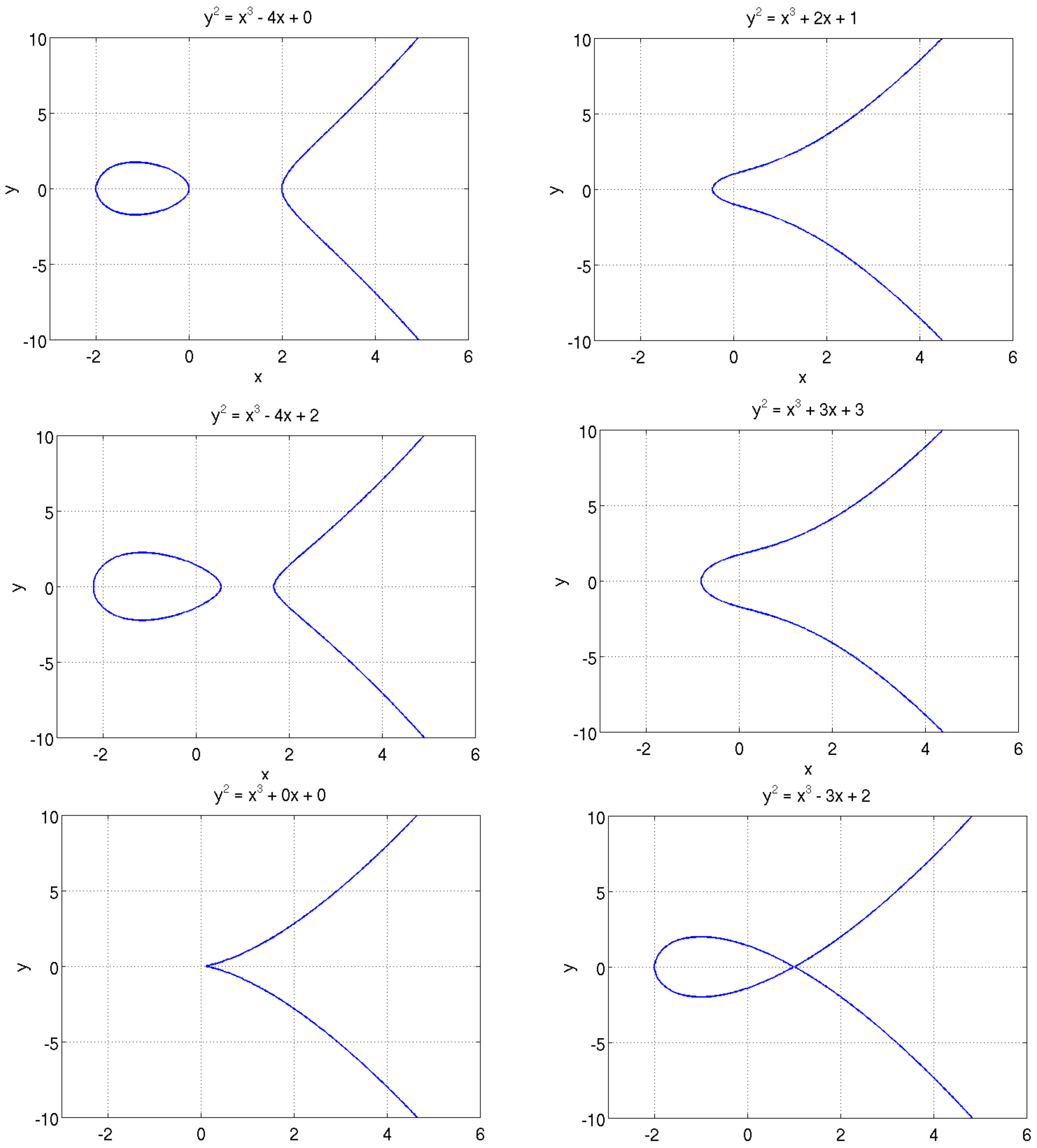


Figure 1: *This figure is from Lecture 14 of "Lecture Notes on Computer and Network Security" by Avi Kak*

14.4: A Group Operator Defined for Points on an Elliptic Curve

- The points on an elliptic curve can be shown to constitute a group.

- Recall from Lecture 4 that a group needs the following: (1) a group operator; (2) an identity element with respect to the operator; (3) closure and associativity with respect to the operator; and (4) the existence of inverses with respect to the operator.

- The group operator for the points on an elliptic curve is, by convention, called **addition**. Its definition has nothing to do with the conventional arithmetic addition.

- To add a point P on an elliptic curve to another point Q on the same curve, we use the following rule

  - We first join P with Q with a straight line. The third point of the intersection of this straight line with the curve, if such an intersection exists, is denoted R. The mirror image of this point with respect to the x-coordinate is the point $P + Q$. If the third point of intersection does **not** exist, we say it is at **infinity**.

- The upper two curves in Figure 2 illustrate the addition operation for two different elliptic curves. The values for a and b for the upper curve at the left are -4 and 0, respectively. The values for the same two constants for the upper curve on the right are 2 and 1, respectively.

- But what happens when the intersection of P and Q is at infinity?

- We denote the point at infinity by the special symbol **O** and we then show that this can serve as the additive identity element for the group operator.

- We now stipulate that $P + \mathbf{O} = P$ for any point on the curve.

- We define the additive inverse of a point P as its mirror reflection with respect to the x coordinate. So if Q on the curve is the mirror reflection of P on the curve, then $Q = -P$. For any such two points, it would obviously be the case that the third point of intersection with the curve of a line passing through the first two points will be at infinity. That is, the point of intersection of a point and its additive inverse will be the distinguished point $\mathbf{O}$.

- We will further stipulate that that $\mathbf{O} + \mathbf{O} = \mathbf{O}$, implying that $-\mathbf{O} = \mathbf{O}$. Therefore, the mirror reflection of the point at infinity is the same point at infinity.

- Now we can go back to the issue of what happens to $P+Q$ when the intersection of two points P and Q is at infinity, as would be the case when P and Q are each other's mirror reflections with regard to the x-axis. Obviously, in this case, the intersection of P and Q is at the distinguished point $\mathbf{O}$, whose mirror reflection is also at $\mathbf{O}$. Therefore, for such points, $P + Q = \mathbf{O}$ and $Q = -P$.

- We have already defined the additive inverse of a point P as its mirror reflection about the x-axis. What is the additive inverse of a point where the tangent is parallel to the y-axis? The additive inverse of such a point is the point itself. That is, if the tangent at P is parallel to the y-axis, then $P + P = \mathbf{O}$.

- In general, what does it mean to add P to itself? To see what it means, let's consider two distinct points P and Q and let Q approach P. The line joining P and Q will obviously become a tangent at P in the limit. Therefore, the operation $P + P$ means that we must draw a tangent at P, find the intersection of the tangent with the curve, and then take the mirror reflection of the intersection.

- For an elliptic curve

$$y^2 \quad = \quad x^3 \;+\; ax \;+\; b$$

we define the set of all points on the curve **along with the distinguished point O** by $E(a, b)$.

- $E(a, b)$ is a group with the "addition" operator as we have defined so far in this section.

- $E(a, b)$ is obviously closed with respect to the addition operation. We can also show geometrically that the property of associativity is satisfied. Every element in the set obviously has its additive inverse in the set.

- Since the operation of "addition" is commutative, $E(a, b)$ is an **abelian group**. (Lecture 4 defines abelian groups.)

- Just for notational convenience, we now define multiplication on this group as repeated addition. Therefore,

$$k \times P \quad = \quad P \ + \ P \ + \ \ldots \ + \ P$$

  with P making k appearances on the right.

- Therefore, we can express $P \ + \ P$ as $2P$, $P \ + \ P \ + \ P$ as $3P$, and so on.

- The two curves at the bottom in Figure 2 show us calculating $2P$ and $3P$ for a given P. The values of a and b for the lower curve on the left are -4 and 2, respectively. The values for the same two constants for the lower curve on the right are both 3.

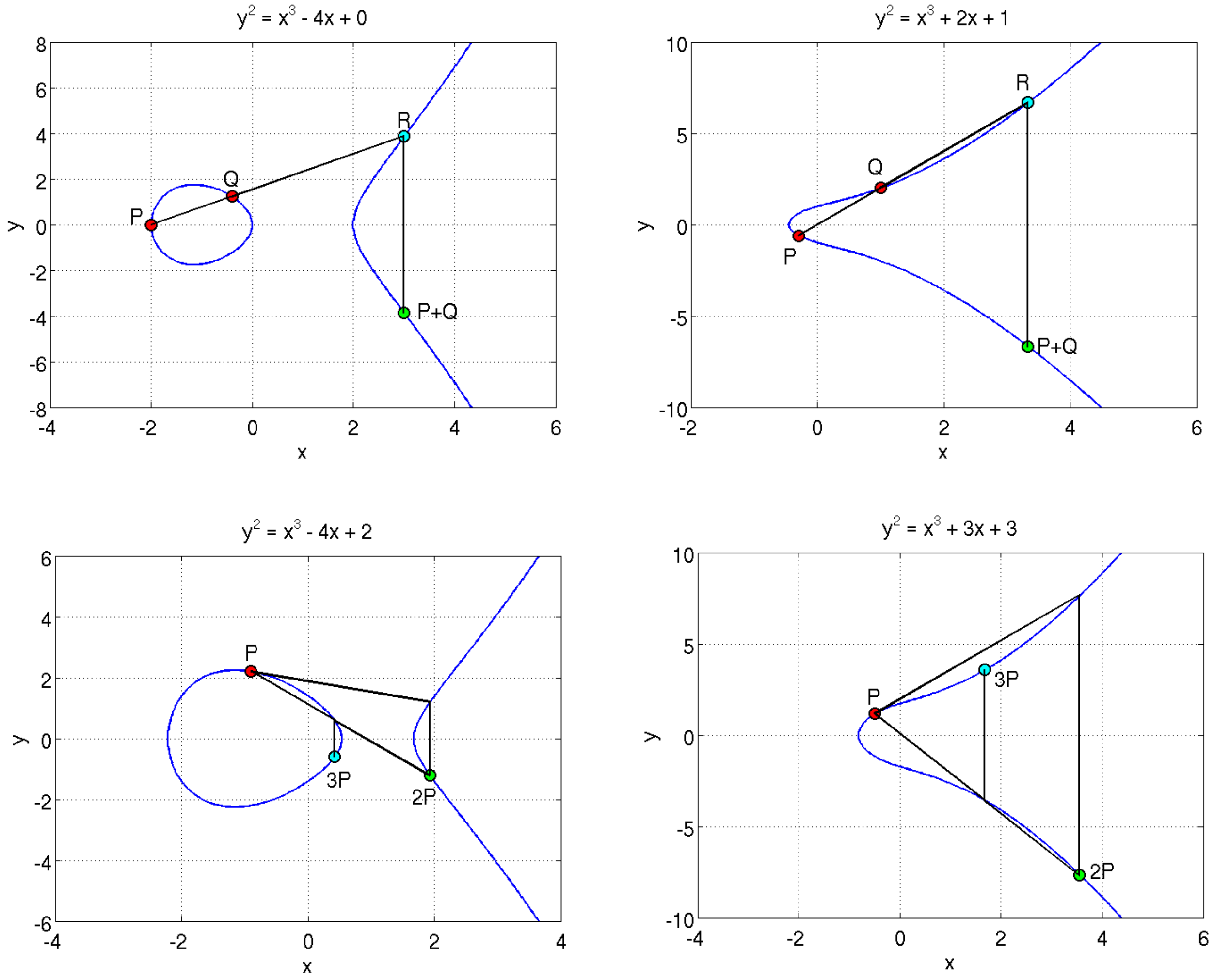


Figure 2: *This figure is from Lecture 14 of "Lecture Notes on Computer and Network Security" by Avi Kak*

14.5: The Characteristic of the Underlying Field and the Singular Elliptic Curves

- The examples of the elliptic curves shown so far were for **the field of real numbers**. (See Lecture 4 for what is meant by a field.) These fields are of **characteristic** zero because no matter how many times you add the multiplicative identity element to itself, you'll never get the additive identity element. (See Section 14.3 for what is meant by the characteristic of a field.)

- The group law of Section 14.4 can also be defined when the underlying field is of characteristic 2 or 3. [As already mentioned in the red and blue explanatory material on page 6, when we consider real numbers modulo 2, we have an underlying field of characteristic 2. By the same token, when we consider real numbers modulo 3, we have an underlying field of characteristic 3.] But now the elliptic curve $y^2 = x^3 + ax + b$ becomes **singular**, a notion that we will define more precisely shortly. While singular elliptic curves do admit group laws of the sort we showed in Section 14.4, such groups, although defined over the points on the elliptic curve, become **isomorphic** to either the multiplicative or the additive group over the underlying field itself, depending on the type of singularity. **That fact makes singular elliptic curves unsuitable for cryptography because they are easy to crack.**

- To show that the elliptic curve $y^2 = x^3+ax+b$ becomes **singular** when the characteristic of the underlying field is 2, let's look at the partial derivatives of the two sides of the equation of this curve:

$$2ydy \quad = \quad 3x^2dx \ + \ adx$$

  implying

$$\frac{dy}{dx} \quad = \quad \frac{3x^2 \ + \ a}{2y} \tag{2}$$

- A point on the curve is **singular** if $\frac{dy}{dx}$ is not properly defined there and a curve that contains a singular point is a **singular curve**. [If $\frac{dy}{dx}$ is not properly defined at a point, then we can construct a tangent at that point. Obviously, such a point would not lend itself to the group law presented in Section 14.4, since that law requires us to draw tangents.] This would be the point where both the numerator and the denominator are zero. [When only the denominator goes to zero, the slope is still defined even though it is ∞.] So the elliptic curve $y^2 = x^3+ax+b$ will become singular if it contains a point (x, y) so that

$$\begin{aligned} 3x^2 \ + \ a \quad &= \quad 0 \\ 2y \quad &= \quad 0 \end{aligned}$$

and the point (x, y) satisfying these two equations lies on the curve.

- Let's now consider the case when the underlying field is of characteristic 2. In this case, we go back to Equation (2) above and see that, since 2 is the same thing as 0 for such a field $\big[$this is based on the definition of **characteristic** in Section 14.3$\big]$, the derivative $\frac{dy}{dx}$ will **not** be defined at $x = \sqrt{\frac{-a}{3}}$. Therefore, the curve $y^2 = x^3 + ax + b$ will be singular for some values of a that can be obtained by substituting $x = \sqrt{\frac{-a}{3}}$ in the equation of the curve.

- Let's now consider the case of a field of characteristic 3. In this case, since 3 is the same thing as 0, we can write for the curve slope from Equation (2):

$$\frac{dy}{dx} \quad = \quad \frac{a}{2y}$$

  This curve becomes singular if we should choose $a = 0$.

- In general, when using the elliptic curve equation $y^2 = x^3 + ax + b$, we avoid underlying fields of characteristic 2 or 3 because of the nature of the constraints they place on the parameters a and b in order for the curve to not become singular.

14.6: An Algebraic Expression for Adding Two Points on An Elliptic Curve

- Given two points P and Q on an elliptic curve $E(a, b)$, we have already pointed out that to compute the point $P + Q$, we first draw a straight line through P and Q. We next find the third intersection of this line with the elliptic curve. We denote this point of intersection by R. Then $P + Q$ is equal to the mirror reflection of R about the x-axis.

- In other words, if P, Q, and R are the three intersections of the straight line with the curve, then

$$P \quad + \quad Q \quad = \quad -R$$

- This implies that the three intersections of a straight line with the elliptic curve must satisfy

$$P \quad + \quad Q \quad + \quad R \quad = \quad \mathbf{O}$$

- We will next examine the algebraic implications of the above relationship between the three points of intersection.

- The equation of the straight line that runs through the points P and Q is obviously of the form:

$$y \quad = \quad \alpha x \ + \ \beta$$

  where α is the slope of the line, which is given by

$$\alpha \quad = \quad \frac{y_Q \ - \ y_P}{x_Q \ - \ x_P}$$

- For a point $(x, \ y)$ to lie at the intersection of the straight line and the elliptic curve $E(a, b)$, the following equality must obviously hold

$$(\alpha x \ + \ \beta)^2 \quad = \quad x^3 \ + \ ax \ + \ b \tag{3}$$

  since $y \ = \ \alpha x + \beta$ on the straight line through the points P and Q and since the equation of the elliptic curve is $y^2 \ = \ x^3 + ax + b$.

- For there to be three points of intersection between the straight line and the elliptic curve, the cubic form in Equation (3) must obviously have three roots. **We already know two of these roots, since they must be x_P and x_Q, correspond to the points P and Q.**

- Being a cubic equation, since Equation (3) has at most three roots, the remaining root must be x_R, the x-coordinate of the third point R.

- Equation (3) represents a **monic polynomial**. What that means is that the coefficient of the highest power of x is 1.

- **A property of monic polynomials is that the sum of their roots is equal to the negative of the coefficient of the second highest power.** Expressing Equation (3) in the following form:

  $$x^3 - \alpha^2 x^2 + (a - 2\alpha\beta)x + (b - \beta^2) = 0 \quad (4)$$

  we notice that the coefficient of x^2 is $-\alpha^2$. Therefore, we have

  $$x_P + x_Q + x_R = \alpha^2$$

  We therefore have the following result for the x-coordinate of R:

  $$x_R = \alpha^2 - x_P - x_Q \quad (5)$$

- Since the point (x_R, y_R) must be on the straight line $y = \alpha x + \beta$, we can write for y_R:

$$\begin{aligned} y_R &= \alpha x_R + \beta \\ &= \alpha x_R + (y_P - \alpha x_P) \\ &= \alpha(x_R - x_P) + y_P \qquad (6) \end{aligned}$$

- To summarize, ordinarily a straight line will intersect an elliptical curve at three points. If the coordinates of the first two points are (x_P, y_P) and (x_Q, y_Q), then the coordinates of the third point are

$$x_R = \alpha^2 - x_P - x_Q \qquad (7)$$

$$y_R = \alpha(x_R - x_P) + y_P \qquad (8)$$

- We started out with the following relationship between P, Q, and R

$$P + Q = -R$$

we can therefore write the following expressions for the x and the y coordinates of the addition of two points P and Q:

$$
\begin{aligned}
x_{P+Q} &= \alpha^2 - x_P - x_Q && (9)\\
y_{P+Q} &= -y_P + \alpha(x_P - x_R) && (10)
\end{aligned}
$$

since the y-coordinate of the reflection $-R$ is negative of the y-coordinate of the point R on the intersecting straight line.

14.7: An Algebraic Expression for Calculating $2P$ from P

- Given a point P on the elliptical curve $E(a, b)$, computing $2P$ (which is the same thing as computing $P + P$), requires us to draw a tangent at P and to find the intersection of this tangent with the curve. The reflection of this intersection about the x-axis is then the value of $2P$.

- Given the equation of the elliptical curve $y^2 = x^3 + ax + b$, the slope of the tangent at a point (x, y) is obtained by differentiating both sides of the curve equation

$$2y\frac{dy}{dx} = 3x^2 + a$$

- We can therefore write the following expression for the slope of the tangent at point P:

$$\alpha = \frac{3x_P^2 + a}{2y_P} \tag{11}$$

- Since drawing the tangent at P is the limiting case of drawing a line through P and Q as Q approaches P, two of the three roots of the following equation (which is the same as Equation (3) you saw before):

$$(\alpha x + \beta)^2 = x^3 + ax + b \tag{12}$$

  must coalesce into the point x_P and the third root must be x_R. As before, R is the point of intersection of the tangent with the elliptical curve.

- As before, we can use the property that sum of the roots of the monic polynomial above must equal the negative of the coefficient of the second highest power. Noting two of the three roots have coalesced into x_P, we get

$$x_P + x_P + x_R = \alpha^2$$

- Substituting the value of α from Equation (11) in the above equation, we get

$$x_R = \alpha^2 - 2x_P = \left(\frac{3x_P^2 + a}{2y_P}\right)^2 - 2x_P \tag{13}$$

- Since the point R must also lie on the straight line $y = \alpha x + \beta$, substituting the expression for x_R in this equation yields

$$\begin{aligned} y_R &= \alpha x_R + \beta \\ &= \alpha x_R + (y_P - \alpha x_P) \\ &= \alpha(x_R - x_P) + y_P \\ &= \frac{3x_P^2 + a}{2y_P}(x_R - x_P) + y_P \end{aligned} \tag{14}$$

- To summarize, if we draw a tangent at point P to an elliptical curve, the tangent will intersect the curve at a point R whose coordinates are given by

$$\begin{aligned} x_R &= \left(\frac{3x_P^2 + a}{2y_P}\right)^2 - 2x_P \\ y_R &= \frac{3x_P^2 + a}{2y_P}(x_R - x_P) + y_P \end{aligned} \tag{15}$$

- Since the value of $2P$ is the reflection of the point R about the x-axis, the value of $2P$ is obtained by taking the negative of the y-coordinate:

$$
\begin{aligned}
x_{2P} &= \left(\frac{3x_P^2 + a}{2y_P}\right)^2 - 2x_P \\
y_{2P} &= \frac{3x_P^2 + a}{2y_P}(x_P - x_R) - y_P
\end{aligned} \tag{16}
$$

14.8: Elliptic Curves Over Z_p for Prime p

- The elliptic curve arithmetic we described so far was over **real numbers**. These curves cannot be used as such for cryptography because calculations with real numbers are prone to round-off error. **Cryptography requires error-free arithmetic.** That is after all the main reason for why we introduced the notion of a finite field in Lectures 4 through 7.

- However, by restricting the values of the parameters a and b, the value of the independent variable x, and the value of the dependent variable y to belong to the **prime finite field** Z_p, we obtain elliptic curves that are more appropriate for cryptography:

$$y^2 \quad \equiv \quad (x^3 \quad + \quad ax \quad + \quad b) \quad (mod\, p) \tag{17}$$

subject to the modulo p version of the same smoothness constraint on the discriminant as we had for the case of real numbers [see Equation (1) in Section 14.3]:

$$(4a^3 \quad + \quad 27b^2) \quad \neq \quad 0 \quad (mod\, p)$$

- We will use the notation $E_p(a, b)$ to represent all the points (x, y) that obey the above equation. $E_p(a, b)$ will also include the distinguished point $\mathbf{O}$, the point at infinity.

- So the points in $E_p(a, b)$ are the set of coordinates (x, y), with $x, y \in Z_p$, such that the equation $y^2 = x^3 + ax + b$, with $a, b \in Z_p$ is satisfied modulo p and such that the condition $4a^3 + 27b^2 \neq 0 \ (mod\ p)$ is fulfilled.

- Obviously, then, the set of points in $E_p(a, b)$ is no longer a curve, but a collection of discrete points in the (x, y) plane (or, even more precisely speaking, in the plane corresponding to the Cartesian product $Z_p \times Z_p$).

- Since the points in $E_p(a, b)$ can no longer be connected to form a smooth curve, we cannot use the geometrical construction to illustrate the action of the group operator. That is, given a point P, now one cannot show geometrically how to compute $2P$, or given two points P and Q, one cannot show geometrically how to determine $P + Q$. **However, the algebraic expressions we derived for these operations continue to hold good provided the calculations are carried out modulo p.**

- Note that for a **prime finite field** Z_p, the value of p is its **characteristic**. (See Section 14.3 for what is meant by the characteristic of a ring.) Elliptic curves over **prime finite fields** with $p \leq 3$, while admitting the group law, are **not** suitable for cryptography. (See Section 14.5)

- The set $E_p(a,b)$ of points, with the elliptic curve defined over a prime finite field Z_p, constitutes a group, the group operator being as defined in Sections 14.6 and 14.7. [In the hierarchy of algebraic structures presented in Lecture 4, the set $E_p(a,b)$ is NOT even a ring since we have not defined multiplication over the set. Yes, we can compute things like $k \times G$ for an element $G \in E_p(a,b)$, since we can construe such a product as repeated addition of the element G, we nevertheless cannot compute a product of arbitrary two elements in $E_{2^n}(a,b)$.]

- We should also mention that you can also define an elliptic curve when the coordinates are drawn from the group $(Z/pZ)^{\times}$ for any positive integer p. The notation $(Z/pZ)^{\times}$ was presented in Section 11.8 of Lecture 11; it consists of the set of all integers that are coprime to p with the group operator being integer multiplication modulo p. In Section 14.14, we will show how an elliptic curve whose points are drawn from $(Z/pZ)^{\times}$ is used in Digital Rights Management. The set $E_p(a,b)$ of points, with the elliptic curve defined over the group $(Z/pZ)^{\times}$, also constitutes a group for the same reasons as stated above.

- As we will see in the next section, elliptic curves can also be defined over **Galois Fields** $GF(2^m)$ that we introduced in Lecture 7. As mentioned in Lecture 7, these are also commonly denoted Z_{2^m} and also commonly called **binary finite fields**. Binary finite fields have characteristic 2. Because of that fact, elliptic curves over $GF(2^m)$ require a form that is different from the one you have seen so far.

14.9: Elliptic Curves Over Galois Fields $GF(2^m)$

- For hardware implementations of ECC, it is common to define elliptic curves over a Galois Field $GF(2^n)$.

- What makes the binary finite fields more convenient for hardware implementations is that the elements of $GF(2^n)$ can be represented by n-bit binary code words. (See Lecture 7.)

- You will recall from Lecture 7 that the addition operation in $GF(2^n)$ is like the XOR operation on bit patterns. That is $x + x = 0$ for all $x \in GF(2^n)$. This implies that a finite field of the form $GF(2^n)$ is of **characteristic** 2. (See Section 14.3 for what is meant by the **characteristic** of a field.)

- As mentioned earlier, the elliptic curve we showed earlier ($y^2 = x^3 + ax + b$) is meant to be used only when the underlying finite field is of characteristic **greater** than 3. (See Section 14.5)

- The elliptic curve equation to use when the underlying field is described by $GF(2^n)$ is

$$y^2 \ + \ xy \ = \ x^3 \ + \ ax^2 \ + b, \qquad b \neq 0 \tag{18}$$

  The constraint $b \neq 0$ serves the same purpose here that the constraint $4a^3 + 27b^2 \neq 0$ did for the case of the elliptic curve equation $y^2 = x^3 + ax + b$. The reason for the constraint $b \neq 0$ is that the discriminant becomes 0 when $b = 0$. As mentioned earlier, when the discriminant becomes zero, we have multiple roots at the same point, causing the derivative of the curve to become ill-defined at that point. In other words, the curve has a singularity at the point where discriminant is 0.

- Shown in Figure 3 are six elliptic curves described by the analytical form $y^2 + xy \ = \ x^3 + ax^2 + b$ for different values of the parameters a and b. The four upper curves are non-singular. The parameters a and b for the top-left curve are 2 and 1, respectively. The same parameters for the top-right curve are 2 and -1, respectively. For the two non-singular curves in the middle row, the one on the left has 0 and 2 for its a and b parameters, whereas the one on the right has -3 and 2. The two curves in the bottom row are both singular, but for different reasons. The one on the left is singular because b is set to 0. As the next section will show, this is a sufficient condition for the discriminant of an elliptic curve (of the kind being studied in this section) to be singular. However, as the next section explains, it is possible for the discriminant of such curves to be singular even when b is not zero. This is demonstrated by the curve on the right in the bottom row.

- The fact that the equation of the elliptic curve is different when the underlying field is $GF(2^n)$ introduces the following changes in the behavior of the group operator:

  - Given a point $P = (x, y)$, we now consider the negative of this point to be located at $-P = (x, -(x+y))$.

  - Given two distinct points $P = (x_P, y_P)$ and $Q = (x_Q, y_Q)$, the addition of the two points, represented by (x_{P+Q}, y_{P+Q}), is now given by

$$\begin{aligned} x_{P+Q} &= \alpha^2 + \alpha - x_P - x_Q - a \\ y_{P+Q} &= -\alpha(x_{P+Q} - x_P) - x_{P+Q} - y_P \end{aligned} \tag{19}$$

with

$$\alpha = \frac{y_Q - y_P}{x_Q - x_P} \tag{20}$$

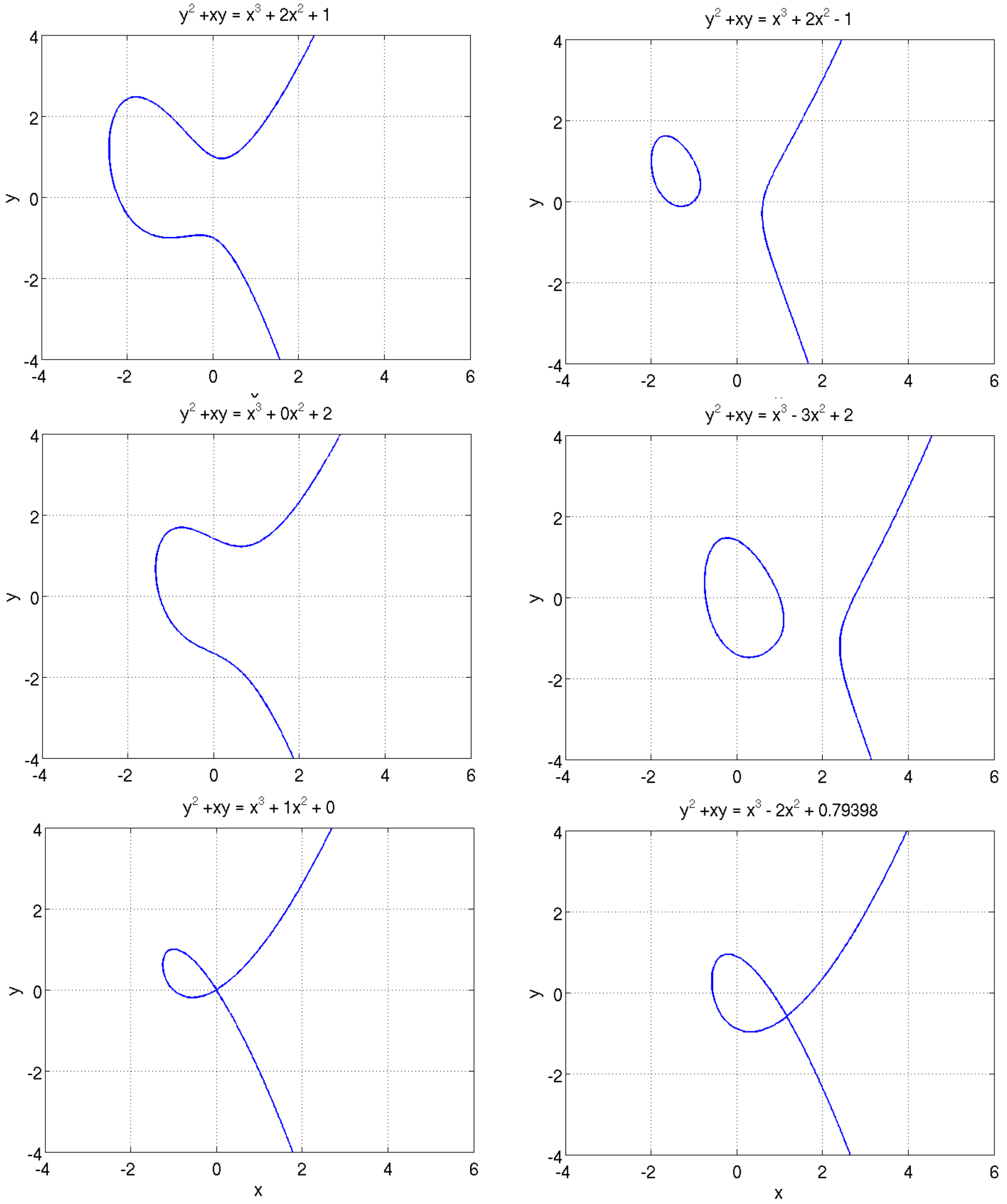


Figure 3: *This figure is from Lecture 14 of "Lecture Notes on Computer and Network Security" by Avi Kak*

– To double a point, that is to calculate $2P$ from P, we now use the formulas

$$\begin{aligned} x_{2P} &= \alpha^2 + \alpha - a - 2x_P \\ y_{2P} &= -\alpha^2 - \alpha + a + (2+\alpha)x_P - \alpha x_{2P} - y_P \end{aligned} \quad (21)$$

with

$$\alpha = \frac{3{x_P}^2 + 2ax_P - y_P}{2y_P + x_P} \quad (22)$$

This value of α is obtained by differentiating both sides of $y^2 + xy = x^3 + ax^2 + b$ with respect to x and writing down an expression for $\frac{dy}{dx}$ just as we derived the expression for α in Equation (11) in Section 14.7.

– Since the results for doubling shown in Equation (21) *can* be obtained (although the style of derivation shown in Section 14.7 is to be preferred) from those in Equation (19) by letting x_Q approach x_P, which in our case can be simply accomplished by setting $x_Q = x_P$, the reader may be puzzled by the very different appearances of the expressions shown for y_{P+Q} and y_{2P}. If you set $x_Q = x_P$ in the expression for y_{P+Q}, then both the y-coordinate expressions can be shown to reduce to $-\alpha^3 - 2\alpha^2 + \alpha(3x_P + a - 1) + 2x_P + a - y_P$.

[The expressions shown in Equations (19) through (22) are derived in a manner that is completely analogous to the derivation presented in Sections 14.6 and 14.7. As before, we recognize that the points on a straight line passing through two points (x_P, y_P) and (x_Q, y_Q) are given by $y = \alpha x + \beta$ with $\alpha = \frac{y_Q - y_P}{x_Q - x_P}$. To find the point of intersection of such a line with the elliptic curve $y^2 + xy = x^3 + ax^2 + b$, as before we form the equation

$$(\alpha x + \beta)^2 + x(\alpha x + \beta) = x^3 + ax^2 + b \tag{23}$$

which can be expressed in the following form as a monic polynomial:

$$x^3 + (a - \alpha^2 - \alpha)x^2 + (-2\alpha\beta - \beta)x + (b - \beta^2) = 0 \tag{24}$$

Reasoning as before, this cubic equation can have at most three roots, of which two are already known, those being the points P and Q. The remaining root, if its exists, must correspond to the point to the point R, which the point where the straight line passing through P and Q meets the curve again. Again using the property that the sum of the the roots is equal to the negative of the coefficient of the second highest power, we can write

$$x_P + x_Q + x_R = \alpha^2 + \alpha - a$$

We therefore have the following result for the x-coordinate of R:

$$x_R = \alpha^2 + \alpha - a - x_P - x_Q \tag{25}$$

Since this point must be on the straight line $y = \alpha x + \beta$, we get for the y-coordinate at the point of intersection $y_R = \alpha x_R + \beta$. Substituting for β from the equation $y_P = \alpha x_P + \beta$, we get the following result for y_R:

$$y_R = \alpha(x_R - x_P) + y_P \tag{26}$$

Earlier we stated that for the elliptic curves of interest to us in this section, the negative of a point $R = (x_R, y_R)$ is given by $-R = (x_R, -(x_R + y_R))$. Since the point (x_{P+Q}, y_{P+Q}) is located at the negative of the point R at (x_R, y_R), we can write the following result for the summation of the two points P and Q:

$$\begin{array}{lcccl} x_{P+Q} & = & x_R & = & \alpha^2 + \alpha - x_P - x_Q - a \\ y_{P+Q} & = & -(x_R + y_R) & = & -\alpha(x_{P+Q} - x_P) + x_{P+Q} - y_P \end{array} \tag{27}$$

The result for doubling of a point can be derived in a similar manner.

Figure 4 shows these operations in action. The two figures in the topmost row show us calculating $P + Q$ for the two points P and Q as shown. The figure on the left in the middle row shows the doubling of a point and the figure on the right the tripling of a point. Shown in the bottom row are the operations of doubling and tripling a point.]

- We will use the notation $E_{2^n}(a,b)$ to denote the set of all points $(x,y) \in GF(2^n) \times GF(2^n)$, that satisfy the equation

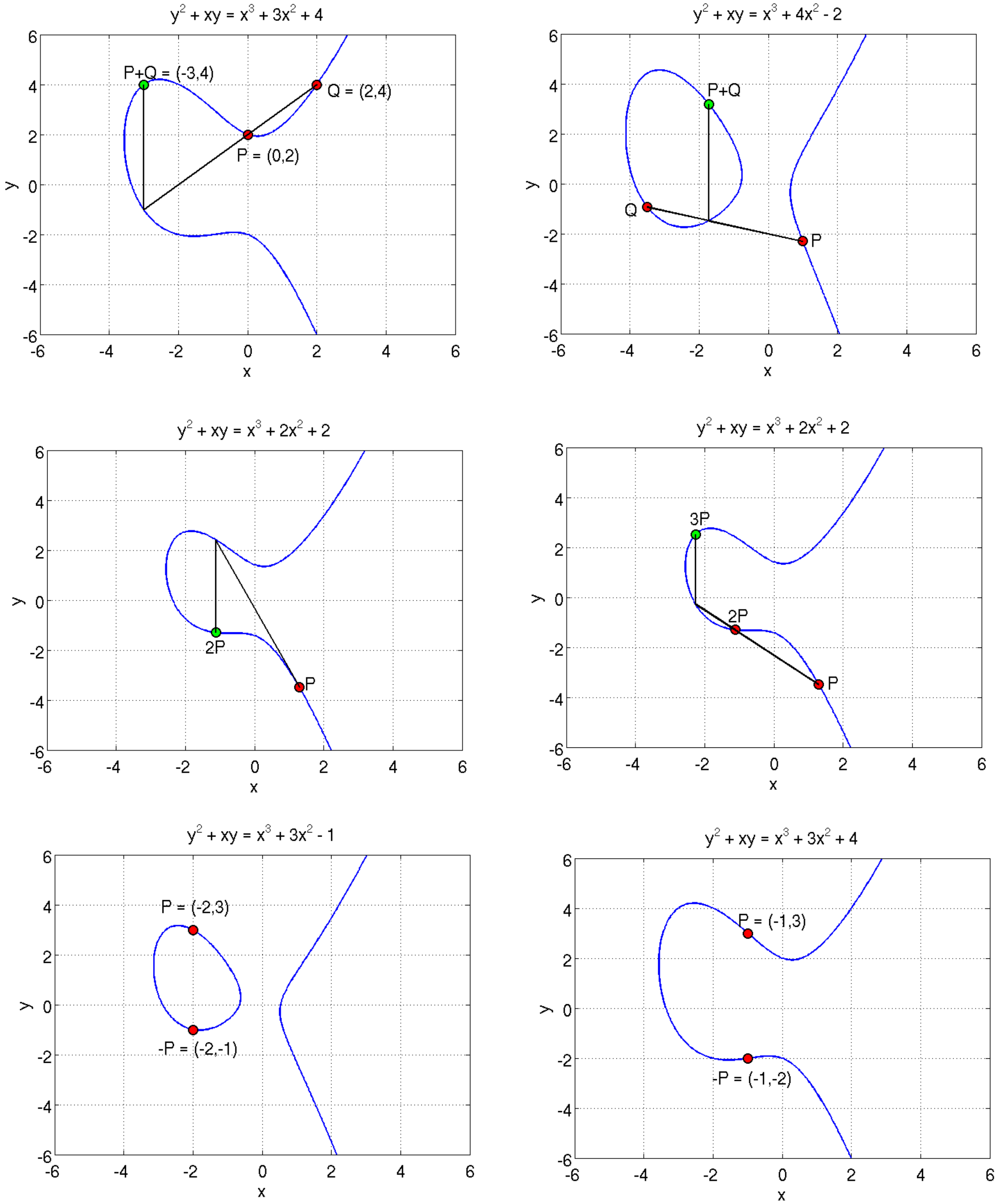
$$y^2 \;+\; xy \;=\; x^3 \;+\; ax^2 \;+\; b,$$

with $a \in GF(2^n)$ and $b \in GF(2^n)$, along with the distinguished point $\mathbf{O}$ that serves as the additive identity element for the group structure formed by the points on the curve. Note that we do not allow b in the above equation to take on the value which is the additive identity element of the finite field $GF(2^n)$.

- If g is a generator for the field $GF(2^n)$ (see Section 7.12 of Lecture 7 for what is meant by the generator of a finite field), then all the element of $GF(2^n)$ can be expressed in the following form

$$0,\ 1,\ g,\ g^2,\ g^3, ..., g^{2^n-2}$$

This implies that the majority of the points on the elliptic curve $E_{2^n}(a,b)$ can be expressed in the form $(g^i,\ g^j)$, where $i,j \;=\; 0,\ 1,\ \ldots,\ n-2$. In addition, there may be points whose coordinates can be expressed $(0,g^i)$ or $(g^i,0)$, with



$y^2 + xy = x^3 + 3x^2 + 4$

$y^2 + xy = x^3 + 4x^2 - 2$

$y^2 + xy = x^3 + 2x^2 + 2$

$y^2 + xy = x^3 + 2x^2 + 2$

$y^2 + xy = x^3 + 3x^2 - 1$

$y^2 + xy = x^3 + 3x^2 + 4$

Figure 4: *This figure is from Lecture 14 of "Lecture Notes on Computer and Network Security" by Avi Kak*

$i = 0, 1, \ldots, n-2$. And then there is, of course, the distinguished point **O**.

- The **order of an elliptic curve**, that is the number of points in the group $E_{2^n}(a,b)$ **is important from the standpoint of the cryptographic security of the curve.** [Note: When we talk about the order of $E_{2^n}(a,b)$, we must of course include the distinguished point **O**.]

- Hasse's Theorem addresses the question of how many points are on an elliptic curve that is defined over a **finite** field. This theorem says that if N is the number of points on $E_q(a,b)$ when the curve is defined on a finite field Z_q with q elements, then N is bounded by

$$|N \; - \; (q \; + \; 1)| \quad \leq \quad 2\sqrt{q}$$

As mentioned previously, N includes the additive identity element **O**.

- Since the Galois field $GF(2^n)$ contains 2^n elements, we can say that the **order** of $E_{2^n}(a,b)$ is equal to $2^n \; + \; 1 \; - \; t$ where t is a number such that $|t| \leq \sqrt{2^n}$.

- An elliptic curve defined over a Galois Field $GF(2^n)$ is **supersingular** if $2|t$, that is if 2 is a divisor of t. [Supersingularity is **not** to be confused with singularity. As previously explained in Section 14.5, when an elliptic curve is defined over real numbers, singularity of the curve is related to its smoothness. More specifically, a curve is singular if its slope at a point is not defined. **Supersingularity**, on the other hand, is related to the order of E_{2^n} and how this order relates to the number of points in the underlying finite field.]

- Should it happen that $t = 0$, then the order of E_{2^n} is $2n + 1$. Since this number is always odd, such a curve can never be supersingular. Supersingular curves defined over fields of characteristic 2 (which includes the binary finite fields $GF(2^n)$) always have an odd number of points, including the distinguished point $\mathbf{O}$.

- Supersingular curves are to be avoided for cryptography because they are vulnerable to the MOV attack. More on that later.

- The set $E_{2^n}(a, b)$ of points constitutes a group, with the group operator as defined by Equations (19) through (22). [In the hierarchy of algebraic structures presented in Lecture 4, the set $E_{2^n}(a, b)$ is NOT even a ring since we have not defined multiplication over the set. Yes, we can compute things like $k \times G$ for an element $G \in E_{2^n}(a, b)$, since we can construe such a product as repeated addition of the element G, we nevertheless cannot compute a product of arbitrary two elements in $E_{2^n}(a, b)$.]

14.10: Is $b \neq 0$ a Sufficient Condition for the Elliptic Curve $y^2 + xy = x^3 + ax^2 + b$ to Not Be Singular?

- In general, we want to avoid using **singular** elliptic curves for cryptography for reasons already indicated.

- In Section 14.9 we indicated that when using a curve of form $y^2 + xy = x^3 + ax^2 + b$, you want to make sure that $b \neq 0$ since otherwise the curve will be singular.

- We will now consider in greater detail when exactly the curve $y^2 + xy = x^3 + ax^2 + b$ becomes singular for the case when the underlying field consists of real numbers. Toward that end we will derive an expression for the discriminant of a polynomial that is singular if and only if the curve $y^2 + xy = x^3 + ax^2 + b$ is singular. The condition which will prevent the discriminant going to zero will be the condition under which the curve $y^2 + xy = x^3 + ax^2 + b$ will stay nonsingular.

- To meet the goal stated above, we will introduce the coordinate transformation

$$y \quad = \quad Y \; - \; \frac{x}{2}$$

in the equation

$$y^2 \;\; + \;\; xy \quad = \quad x^3 \;\; + \;\; ax^2 \;\; + b$$

- The purpose of the coordinate transformation is to get rid of the troublesome term xy in the equation. Note that this coordinate transformation cannot make a singularity disappear, and neither can it introduce a new singularity. With this transformation, the equation of the curve becomes

  $$Y^2 \; - \; \frac{x^2}{4} \quad = \quad x^3 \;\; + \;\; ax^2 \;\; + \;\; b$$

  which can be rewritten as

  $$Y^2 \quad = \quad x^3 \;\; + \;\; (a + \frac{1}{4})x^2 \;\; + \;\; b$$

  The polynomial on the right hand side of the equation shown above has a singular point wherever its discriminant goes to zero.

- In general, the discriminant of the polynomial

$$a_3 z^3 \quad + \quad a_2 z^2 \quad + \quad a_1 z \quad = \quad 0$$

is given by

$$D_3 \quad = \quad a_1^2 a_2^2 \; - \; 4a_0 a_2^3 \; - \; 4a_1^3 a_3 \; + \; 18a_0 a_1 a_2 a_3 \; - \; 27a_0^2 a_3^2$$

- Substituting the coefficient values for our case, $a_3 = 1$, $a_2 = (a + \frac{1}{4})$, $a_1 = 0$, and $a_0 = b$, in the general formula for the discriminant of a cubic polynomial, we get for the discriminant

$$D_3 \quad = \quad -4b \left(a + \frac{1}{4} \right)^3 \quad - \quad 27b^2$$

This simplifies to

$$D_3 \quad = \quad \frac{1}{16} \left[-64a^3 b \; - \; 48a^2 b \; - \; 12ab \; - \; b \; - \; 432b^2 \right]$$

which can be expressed as

$$D_3 \quad = \quad -\frac{1}{16} b \left[64a^3 \; + \; 48a^2 \; + \; 12a \; + \; 432b \; + \; 1 \right]$$

- Obviously, if $b = 0$, the discriminant will become 0. However, it is also obvious that even when the $b = 0$ condition is satisfied, certain values of a and b may cause the discriminant to go to 0.

- As with the supersingular curves, elliptic curves that are singular are to be avoided for cryptography because they are vulnerable to the MOV attack.

14.11: Elliptic Curve Cryptography — The Basic Idea

- That elliptic curves over finite fields could be used for cryptography was suggested independently by Neal Koblitz (University of Washington) and Victor Miller (IBM) in 1985.

- Just as RSA uses multiplication as its basic arithmetic operation (exponentiation is merely repeated multiplication), ECC uses the "addition" group operator as its basic arithmetic operation (multiplication is merely repeated addition).

- Suppose G is a user-chosen "base point" on the curve $E_q(a, b)$, where $q = p$ for some prime p when the underlying finite field is a prime finite field and $q = 2^n$ when the underlying finite field is a Galois field.

- In accordance with how the group operator works, $k \times G$ stands for $G + G + G + \ldots + G$ with G making k appearances in this expression.

- Now suppose our message consists of an integer M and we encrypt it by calculating $C = M \times G$. [For the purpose of visualization, think of $M \times G$ as the two-dimensional point G being added to itself M times through the geometric construction you saw in Section 14.4.] Now the question is whether an adversary with knowledge of all of the parameters of the curve $E_q(a, b)$ and of the point G can decrypt C and figure out the value of the message integer M. [Bear in mind that whereas M is an integer, C just like G is a point on the elliptic curve. In that sense, M and C are two different types of entities.]

- The core notion that ECC is based on is that, with a proper choice for G, whereas it is relatively easy to calculate $C = M \times G$, it can be extremely to recover M from C even when an adversary knows the curve $E_q(a, b)$ and the G used. Recovering M from C is referred to as having to solve the **discrete logarithm** problem. [To understand why finding M from C is referred to as solving the discrete logarithm problem: Note that the word "addition" for the group operator for $E_q(a, b)$ is a matter of convention and convenience. As you already know from Lecture 4, a group operator is typically referred to as addition and denoted '+'. There is obviously nothing wrong with choosing to express $G + G + G + \ldots + G$ more generically as $G \circ G \circ G \circ \ldots \circ G$ if we do not want to get confused by our deeply rooted mental associations with the '+' operator. **Now let's see what we mean by a logarithm.** As you know, if $a = b^n$ then $n = \log_b a$. We are at liberty to write b^n as $b \times b \times b \ldots \times b$, or even as $b \circ b \circ b \ldots \circ b$ if we assume that the operator $\circ$ stands for multiplication. If we want to recover the **number of times** b participates in $a = b \circ b \circ b \ldots \circ b$ we take the logarithm of a to the base b. By the same token, if we want to determine the **number of times** G participates in $C = G \circ G \circ G \circ \ldots \circ G$, we take the "logarithm" of C to the base G.]

- An adversary could try to recover M from $C = M \times G$ by calculating $2G$, $3G$, $4G$, $\ldots$, kG with k spanning the size of the set $E_q(a, b)$, and then seeing which one of the results matched C. But if q is sufficiently large and if the point G on the curve $E_q(a, b)$ is chosen carefully, that would take much too long.

- At this point, the reader may ask: How does the person who has calculated $C = M \times G$ recover M from G, considering that $E_p(a, b)$ is only a group? That is, considering that multiplicative inverses are not directly defined in $E_p(a, b)$, how do we recover M from C without engaging in the extremely difficult task of having to solve the discrete logarithm problem. [The last statement here is meant more for dramatic effect than to convey a technically correct point. To elaborate, recovering M from the product $M \times G$ is NOT an exercise in multiplicative inversion, in the regular sense of what is meant by multiplicative inversion, since the message integer M is NOT an element of $E_p(a, b)$. M is a plain old integer that tells us how many times G should be added to itself in order to form C. Nonetheless, it is good to keep in mind that $E_p(a, b)$ is merely a group and thus has certain limitations.] Properly stated, recovering M from the product $M \times G$ is an exercise in solving the discrete logarithm problem, as explained earlier in this section.

- Fortunately, as the reader will see in the next section, we will not use the products $M \times G$ directly for encryption. Our goal in this section was simply to demonstrate that when you construct a product like $M \times G$, assuming you already know the operand G, it is still computationally very difficult to find the other operand G. In the next section, we will use these products in a Diffie-Hellman based approach to cryptography with elliptic curves.

14.12: Elliptic Curve Diffie-Hellman Secret Key Exchange

- The reader may wish to first review Section 13.5 of Lecture 13 before proceeding further. The Diffie-Hellman idea was first introduced in that section.

- A community of users wishing to engage in secure communications with ECC chooses the parameters q, a, and b for an elliptic-curve based group $E_q(a, b)$, and a base point $G \in E_q(a, b)$.

- A selects an integer X_A to serve as his/her private key. A then generates $Y_A = X_A \times G$ to serve as his/her public key. A makes publicly available the public key Y_A.

- B designates an integer X_B to serve as his/her private key. As was done by A, B also calculates his/her public key by $Y_B = X_B \times G$.

- In order to create a shared secret key (that could subsequently be used for, say, a symmetric-key based communication link), both

A and B now carry out the following operations:

- A calculates the shared session key by

$$K = X_A \times Y_B \tag{28}$$

- B calculates the shared session key by

$$K = X_B \times Y_A \tag{29}$$

- The calculations in Eqs. (19) and (20) yield the same result because

$$\begin{aligned} K \text{ as calculated by } A &= X_A \times Y_B \\ &= X_A \times (X_B \times G) \\ &= (X_A \times X_B) \times G \\ &= (X_B \times X_A) \times G \\ &= X_B \times (X_A \times G) \\ &= X_B \times Y_A \\ &= K \text{ as calculated by } B \end{aligned}$$

- To discover the secret session key, an attacker could try to discover X_A from the publicly available base point G and the publicly available Y_A. Recall, $Y_A = X_A \times G$. But, as already explained in Section 14.11, this requires solving the discrete logarithm problem which, for a properly chosen set of curve parameters and G, can be extremely hard.

- To increase the level of difficulty in solving the discrete logarithm problem, we select for G a base point whose **order** is very large. The **order** of a point on the elliptic curve is the **least number of times** G must be added to itself so that we get the **identity element O** of the group $E_q(a, b)$. [We can also associate the notion of **order** with an elliptic curve over a finite field: The **order of an elliptic curve** is the total number of points in the set $E_q(a, b)$. This order is denoted $\#E_q(a, b)$.

- Since the integers X_A, Y_A, X_B, and Y_B must all be less than the order of the base point G, the value of the order of the base point must also be made publicly available.

- The base point G is also known as the **generator** of a **subgroup** of $E_q(a, b)$ whose elements are all given by G, $2G$, $3G$, ..., and, of course, the identity element **O**. For the size of the subgroup to equal the **degree** of the generator G, the value of n must be a prime when the underlying field is a Galois field

$GF(2^n)$.

14.13: Security of ECC

- Just as RSA depends on the difficulty of large-number factorization for its security, ECC depends on the difficulty of the large number discrete logarithm calculation. This is referred to as the **Elliptic Curve Discrete Logarithm Problem** (ECDLP).

- It was shown by Menezes, Okamoto, and Vanstone (MOV) in 1993 that (for supersingular elliptic curves) the problem of solving the ECDLP problem (where the domain is the group $E_q(a, b)$) can be reduced to the much easier problem of finding logarithms in a finite field. There has been much work recently on extending the MOV reduction to general elliptic curves.

- In order to not fall prey to the MOV attack, the underlying elliptic curve and the base point chosen must satisfy what is known as the **MOV Condition**.

- The MOV condition is stated in terms of the **order** of the base point G. The order m of the base point G is the value of m such that $m \times G = \mathbf{O}$ where $\mathbf{O}$ is the additive identity element of the group $E_q(a, b)$ as defined in Section 14.4.

- The MOV condition states that the **order** m of the base-point should not divide $q^B - 1$ for small B, say for $B < 20$. Note that q is the prime p when the underlying finite field is Z_p or it is 2^n when the underlying finite field is $GF(2^n)$.

- When using $GF(2^n)$ finite fields, another security consideration relates to what is known as the **Weil descent attack**. To not be vulnerable to this attack, n must be a prime.

- Elliptic curves for which the total number of points on the curve equals the number of elements in the underlying finite field are also considered cryptographically weak.

14.14: ECC For Digital Rights Management

- ECC has been and continues to be used for Digital Rights Management (DRM). DRM stands for technologies/algorithms that allow a content provider to impose limitations on the whos and hows of the usage of some media content made available by the provider.

- ECC is used in the DRM associated with the Windows Media framework that is made available by Microsoft to third-party vendors interested in revenue-generating content creation and distribution. In what follows, we will refer to this DRM as **WM-DRM**.

- The three main versions of WM-DRM are Version 1 (released in 1999), Version 2 (released in 2003, also referred to as Version 7.x and Version 9), and Version 3 (released in 2003, also known as Version 10). All three versions have been cracked. As you would expect in this day and age, someone figures out how to strip away the DRM protection associated with, say, a movie and makes both the unprotected movie and the protection stripping algorithm available anonymously on the web. In the meantime, the content provider (like Apple, Sony, Microsoft, etc.) comes out with a patch to fix the exploit. Thus continues the cat and mouse game between the big content providers and the anonymous "crackers."

- Again as you would expect, the actual implementation details of most DRM algorithms are proprietary to the content providers and distributors. But, on October 20, 2001, an individual, under the pseudonym Beale Screamer, posted a detailed description of the inner workings of the WM-DRM Version 2. This information is still available at the URLs `http://cryptome.org/ms-drm.htm` and `http://cryptome.org.beale-sci-crypt.htm` where you will find a command-line tool named `FreeMe` for stripping away the DRM protection of the older versions of Windows Media documents. Since Version 2 is now considered out of date, the main usefulness of the information posted at the web site lies in its educational value.

- WM-DRM Version 2 used elliptic curve cryptography for exchanging a secret session key between a user's computer and the license server at the content provider's location. As to how that can be done, you have already seen the algorithm in Section 14.12.

- The ECC used in WM-DRM V. 2 is based on the first elliptic curve $y^2 = x^3 + ax + b$ that was presented in Section 14.3. The ECC algorithm used is based on the points on the curve whose x

and y coordinates are drawn from the finite field $(Z/pZ)^{\times}$, which we defined in Section 14.8, with the number p set to

$$p = 785963102379428822376694789446897396207498568951$$

In the WM-DRM ECC, all are represented using 20 bytes. Here is the hex representation of the modulus p shown above:

$$p = 0x89abcdef012345672718281831415926141424f7$$

- We also need to specify values for the parameters a and b of the elliptic curve $y^2 = x^3 + ax + b$. As you would expect, these parameters are also drawn from $(Z/pZ)^{\times}$ and their values are given by

$$\begin{aligned} a &= 317689081251325503476317476413827693272746955927 \\ b &= 79052896607878758718120572025718535432100651934 \end{aligned}$$

Since all numbers in the ECC implementation under consideration are stored as blocks of 20 bytes, the hex representations of the byte blocks stored for a and b are

$$\begin{aligned} a &= 0x37a5abccd277bce87632ff3d4780c009ebe41497 \\ b &= 0x0dd8dabf725e2f3228e85f1ad78fdedf9328239e \end{aligned}$$

- Following the discussion in Sections 14.11 and 14.12, the ECC algorithm would also need to choose a base point G on the elliptic curve $y^2 = x^3 + ax + b$. The x and the y coordinates of this point in the ECC as implemented in WM-DRM are

$$\begin{aligned} G_x &= 771507216262649826170648268565579889907769254176 \\ G_y &= 390157510246556628525279459266514995562533196655 \end{aligned}$$

The 20-byte hex representations for these two coordinates are

$$\begin{aligned} G_x &= 0x8723947fd6a3a1e53510c07dba38daf0109fa120 \\ G_y &= 0x445744911075522d8c3c5856d4ed7acda379936f \end{aligned}$$

- As mentioned in Section 14.12, an ECC protocol must also make publicly available the order of the base point. For the present case, this order is given by

$$\#E_p(a, b) = 785963102379428822376694789446897396207498568951$$

- With the elliptic curve and its parameters set as above, the next question is how exactly the ECC algorithm is used in WM-DRM.

- When you purchase media content from a Microsoft partner peddling their wares through the Window Media platform, you would need to download a “license” to be able play the content on your computer. Obtaining the license consists of your computer randomly generating a number $n \in Z_p$ for your computer’s private key. Your computer then multiplies the base point G with the private key to obtain the public key. Subsequently your computer can interact with the content provider’s license server in the manner described in Section 14.12 to establish a secret session key for the transfer of license related information into your computer.

- In order to ensure that only your computer can use the downloaded license, WM-DRM makes sure that you cannot access the private key that your computer generated for the ECC algorithm. Obviously, if you could get hold of that n, you could pass the encrypted content file and the private key to your friend and they would be able to pretend to be you vis-a-vis the license server. WM-DRM hides an RC4 encrypted version of the private key in the form of a linked list in which each nodes stores one half of the key.

- When DRM software is cracked, it is usually done by what is known as “hooking” the DRM libraries on a computer as they dump out either the keys or the encrypted content.

HOMEWORK PROBLEMS

1. Why is there all this excitement about Elliptic Curve Cryptography?

2. How do we construct the number system to use for ECC?

3. ECC uses numbers that correspond to points on elliptic curves. What is an elliptic curve? Does it have anything to do with an ellipse?

4. What is the geometrical interpretation of the group law that is used for the numbers drawn from the elliptic curves in ECC?

5. What is the fundamental reason for why ECC can use shorter keys for providing the same level of security as what RSA does with much longer keys?

Acknowledgments

I'd like to thank Helena Verrill and Subhash Kak for sharing their insights with me on the mathematics of elliptic curves and on the subject of elliptic curve cryptography. Helena Verrill is the source of much of the information provided regarding the singularity and supersingularity of elliptic curves.

All of the figures in this version were generated by Chad Aeschliman. My request to Chad was to just help me out with the figures. But, with his aversion to doing anything by halves, he decided to first master the subject of elliptic curves defined over finite fields. The derivation steps shown on page 36 were worked out by Chad. Chad is working on his Ph.D. in the Robot Vision Lab at Purdue.