Math 4527 (Number Theory 2)

Lecture #15 of 38 $\sim$ February 24, 2021

Elliptic Curve Factorization and Cryptography

- Analysis and Examples of Elliptic Curve Factorization
- Elliptic Curve Cryptography $\sim$ Encoding Messages

This material represents §7.2.1-7.2.2 from the course notes.

Analyzing Elliptic Curve Factorization, I

Last time, we discussed Lenstra's algorithm:

Algorithm (Lenstra's Elliptic-Curve Factorization Algorithm)

Suppose n is composite.
Choose a bound M, a point $P = (x_0, y_0)$, and an integer A.
Let E_n be the elliptic curve $y^2 = x^3 + Ax + B$ modulo n with B chosen so that P lies on E.
Set $Q_1 = P$ and for $2 \leq j \leq M$, define $Q_j = jQ_{j-1}$ (on E_n).
If at any stage of the computation the point Q_j cannot be computed, due to a necessary division by a denominator d which is not 0 modulo n but which is not invertible modulo n, then $\gcd(d, n)$ is a proper divisor of n. If a divisor is not found and Q_M is not ∞, increase the value of M and continue the computation. Otherwise, if $Q_M = \infty$, repeat the procedure with a new choice of P and A.

Analyzing Elliptic Curve Factorization, II

A few preliminary remarks:

- The curve E can be singular, as long as P is not the singular point on the curve. (By "singular" we mean singular mod p or mod q, which is equivalent to saying that the discriminant Δ has a common prime divisor with n.)
- However, choosing E to be a singular curve is not optimal, because (as it turns out) the algorithm will essentially reduce either to Pollard's $(p-1)$-algorithm or trial division according to the type of singularity.

Analyzing Elliptic Curve Factorization, III

Example: Use Lenstra's factorization algorithm to find a divisor of the integer $n = 170999$ using the point $P = (1, 4)$ on the elliptic curve $E : y^2 = x^3 + 4x + 11$.

- We simply compute the points Q_j successively using the recursion $Q_1 = P$, $Q_j = jQ_{j-1}$ on the elliptic curve E modulo n until we obtain a problematic denominator.

j	1	2	3	4
Q_j	$(1, 4)$	$(109545, 75144)$	$(81282, 86818)$	$(100818, 143145)$
Factor?	no	no	no	no

j	5	6	7	8
Q_j	$(152033, 116998)$	$(87978, 17295)$	$(104368, 99929)$	$(126411, 167685)$
Factor?	no	no	no	no

j	9	10		
Q_j	$(79623, 108587)$	–		
Factor?	no	557		

Analyzing Elliptic Curve Factorization, IV

Example: Use Lenstra's factorization algorithm to find a divisor of the integer $n = 170999$ using the point $P = (1, 4)$ on the elliptic curve $E : y^2 = x^3 + 4x + 11$.

- In this case, attempting to compute $10Q_9$ will require dividing by a denominator that is not relatively prime to n.
- The exact details of the computation will depend on the method used to compute $10Q_9$, but successive doubling will yield $2Q_9 = (147257, 97701)$ and $8Q_9 = (160625, 116187)$, and attempting to add these two points will require using a line with slope $m = \dfrac{116187 - 97701}{160625 - 147257} = \dfrac{18486}{13368}$, and $\gcd(13368, 170999) = 557$.

Analyzing Elliptic Curve Factorization, V

The elliptic curve factorization algorithm seems to work, but it is not obvious how fast it is nor how efficient it is in comparison to other algorithms.

- The factorization algorithm will succeed after M steps when the order of P on the elliptic curve E_p (i.e., E modulo p) divides $M!$, but the order of P on E_q (i.e., E modulo q) does not divide $M!$.
- It is unlikely that these two things will occur at exactly the same value of M, so what we are really seeking is for the order of P on E_p to divide $M!$.
- From our results on orders, we know that the order of P on E_p divides the number of points N on E_p, so we are certainly guaranteed to succeed if N divides $M!$.
- Thus, the elliptic curve factorization will succeed quickly as long as the prime divisors of N are all fairly small.

Analyzing Elliptic Curve Factorization, VI

Note that this is a similar criterion to that of Pollard's $(p-1)$-algorithm, which succeeds quickly as long as the prime divisors of $p-1$ are all fairly small. (An integer all of whose prime divisors are $\leq M$ is called M-smooth.)

- However, we are free to make different choices for the elliptic curve E, each of which will give a different random integer that is near p. As long as one of the curves we choose is M-smooth, we will obtain the factorization of n.
- By the Hasse bound, $|N - p - 1| \leq 2\sqrt{p}$.
- As we discussed, it is known that N can take any integral value in the Hasse-bound interval, where the values in the center of the interval are the most common.

Analyzing Elliptic Curve Factorization, VII

Since we are free to switch to a different curve in cases where the factorization method would take a long time, we see that elliptic curve factorization is much more versatile than Pollard's $(p-1)$-algorithm.

- Explicitly, if we are using Pollard's $(p-1)$-algorithm, if $p-1$ has a large prime divisor then we are simply out of luck, but with elliptic curve factorization if N has a large prime divisor then we can simply switch to a different curve.
- Of course, we will generally not know the exact value of N, so we would instead switch curves if we have spent a long time computing and not gotten any results yet.
- As a practical matter, what we could do instead is run simultaneous computations on many different elliptic curves, rather than "switching" after we exhaust one computation.

Analyzing Elliptic Curve Factorization, VIII

Another advantage to using several curves is that the computations can be completely parallelized (i.e., they can be run on separate processors), since the point operations on different curves have nothing to do with one another.

- It is a rather nontrivial analytic number theory problem to determine the appropriate heuristic for the density of integers in the Hasse interval $|N - p| \leq 2\sqrt{p}$ that are M-smooth, which is needed in order to estimate how many curves should be used in order to search for the factorization and to estimate the value of M that should be used.
- We will not give the details of this computation, but the approximately optimal pairs (M, k) for the bound M and the number of curves k are roughly $(2000, 25)$ for 15-digit prime divisors, $(10000, 100)$ for 20-digit prime divisors, and $(50000, 300)$ for 25-digit prime divisors.

Analyzing Elliptic Curve Factorization, IX

Overall, if one computes the total time requirement with optimal choices for the parameters, Lenstra's elliptic curve algorithm can factor an integer n in a total of approximately $e^{\sqrt{2}(\ln p)^{1/2}(\ln \ln p)^{1/2}}$ steps, where p is the smallest prime divisor of n.

- This number of steps is bounded above by $e^{(\ln n)^{1/2}(\ln \ln n)^{1/2}}$, and so the elliptic curve factorization has roughly the same asymptotic speed as the quadratic sieve.
- In practice, due to the fact that elliptic curve operations are slower than modular exponentiations, Lenstra's algorithm becomes slower than the sieve methods for integers exceeding 60 digits or so, and is slower than Pollard's ρ-algorithm for numbers under 30 digits.
- However, the elliptic curve method is much more efficient at finding comparatively small divisors (around 30 digits or less) of large integers than the sieve methods are.

Analyzing Elliptic Curve Factorization, X

Many implementations of general-purpose factorization algorithms (e.g., in software systems like Mathematica or Sage) use a combination of different approaches to search for factors of various different sizes.

- A typical setup is to use some combination of trial division, the Pollard $(p-1)$-algorithm, and the Pollard ρ-algorithm to search for small factors (under 15 digits or so).
- Next, use Lenstra's algorithm to search for factors of medium size (15-30 digits).
- Finally, use a sieve method (the quadratic sieve or the general number field sieve) to factor the remaining portion of the integer, which will now be a product only of large primes.

Analyzing Elliptic Curve Factorization, XI

There are several improvements and optimizations that can be made to Lenstra's original algorithm.

- The largest computational overhead in Lenstra's algorithm is computing the point multiplications.
- There are various ways to arrange the arithmetic operations in such a way that fewer computations are needed: in particular, it is possible to use both additions and subtractions when doing successive doubling (since computing the inverse of a point is essentially free).

Analyzing Elliptic Curve Factorization, XII

Furthermore, by using different models for elliptic curves other than the reduced Weierstrass form $y^2 = x^3 + Ax + B$, some additional savings are possible.

- It is also possible to choose the elliptic curve $y^2 = x^3 + Ax + B$ in such a way that it is still essentially random modulo n, but is guaranteed to have a point of some specific small order, such as 12.
- Such restrictions then imply that the number of points on the curve is divisible by 12, which marginally reduces the size of potential large prime divisors of N.

There are some other improvements that can be made as well using ideas from other factorization algorithms (e.g., the Pollard ρ-algorithm).

Some Cryptography Fundamentals, I

Now we will discuss how to use elliptic curves for doing cryptography. A quick introduction to some fundamentals:

- Alice and Bob refer to two parties attempting to exchange information. (Generally, Alice wants to send a message to Bob, though the communication can be two-directional.) Additional parties are usually named with the letters following (Carol, Dave, etc.).
- Eve refers to a non-malicious eavesdropper, who can listen in to the communications between Alice and Bob, but will not alter them.
- Mallory refers to a malicious eavesdropper, who can listen to Alice and Bob's communications and may also attempt to impersonate them or alter their messages.

Some Cryptography Fundamentals, II

Encoding and decoding in our cryptosystems each require some specific piece of information, called a key. In general, the process works as follows:

1. Alice wishes to send a secure message to Bob.
2. Alice takes her unencrypted message, her plaintext, and encrypts it using her encryption key to obtain a ciphertext.
3. Alice then sends the ciphertext to Bob, who then uses his decryption key to decode, thus recovering Alice's original plaintext message.

When we encode actual messages, I will write plaintexts in **bold lowercase** and ciphertexts in **BOLD UPPERCASE**.

Some Cryptography Fundamentals, III

Some cryptosystems are symmetric: the information required to encode a message is the same as the information required to decode a message (i.e., the encoding and decoding keys can be obtained from each other).

- Many historical cryptosystems were symmetric: the (in)famous "Caesar shift", the more general class of alphabet-substitution ciphers, the Vigenère cipher, the Playfair cipher, ADFGX, one-time pads, and so forth.
- Modern examples of symmetric cryptosystems include DES ("Data Encryption Standard") used through the 1980s, and AES ("Advanced Encryption Standard") which is in use today.

Some Cryptography Fundamentals, IV

Other cryptosystems are asymmetric: the information required to decode a message is very different from the information required to encode a message.

- Asymmetric encryption is more modern: examples of asymmetric cryptosystems include RSA, ElGamal, and the elliptic curve cryptosystems we will discuss.
- In most such systems, the encoding and decoding methods are sufficiently distinct that one may publicize the encryption key ("public-key encryption") without worry that this will make easy decryption possible.
- Ultimately, public-key cryptosystems revolve around the existence of so-called one-way functions: functions which are easy to evaluate ("easy forward") but very difficult to invert ("hard backward") on most outputs.

Some Cryptography Fundamentals, V

Ultimately, public-key cryptosystems revolve around the existence of so-called one-way functions: functions which are easy to evaluate but very difficult to invert on most outputs.

- As an example, consider the function $f(p, q) = pq$ that takes two prime numbers and outputs their product.
- It is trivial to compute the product pq given p and q, but if we are given pq and asked to find p and q, we would need to know how to factor an arbitrary integer, which (as we have already discussed) seems to be much harder.
- The property that factorization is much harder than multiplication is the basis for many public-key cryptosystems, including RSA.
- Another problem that is often used is the difficulty of computing discrete logarithms in groups: in $\mathbb{Z}/m\mathbb{Z}$ this means solving $a^n = b \pmod{m}$ for $n = \log_a b$.

Encoding Messages on Elliptic Curves, I

In order to use elliptic curves for cryptography, we must first encode messages as points on elliptic curves.

- With cryptosystems based on modular arithmetic, we can simply write a message as a residue class modulo m (usually with some kind of padding scheme to increase security).
- But it is not quite so trivial to encode a message as a point on an elliptic curve if we specify the curve E ahead of time, as would be necessary for a public-key cryptosystem.

Encoding Messages on Elliptic Curves, II

To see why, suppose we have chosen an elliptic curve $y^2 = x^3 + Ax + B$ modulo a prime p, and wish to convert a message m into a point on the curve.

- We can assume that m is smaller than p, since we may break m up into pieces and send each piece separately using whatever scheme we come up with.
- The issue is that we cannot, for example, simply break a messge in half and write down the point (m_1, m_2), since there is no reason to expect that (m_1, m_2) will lie on the curve $y^2 = x^3 + Ax + B$.
- A more sensible approach would be to place the message in one coordinate. Since taking square roots is easier than solving cubics, we should try looking for a point (m, y) on E.
- But of course, there may not be a value of y satisfying the equation $y^2 = m^3 + Am + B \pmod{p}$, so this could also fail.

Quadratic Residues, I

To handle this issue, we need to recall some results about quadratic residues modulo p.

Definition

If a is a residue class modulo m, we say a is a quadratic residue if there is some b such that $b^2 \equiv a \pmod{m}$. If there is no such b, then we say a is a quadratic nonresidue.

Examples:

1. Mod 5, the quadratic residues are 0, 1, and 4, while the nonresidues are 2 and 3.
2. Mod 7, the quadratic residues are 0, 1, 4, and 2, while the nonresidues are 3, 5, and 6.
3. Mod 13, the quadratic residues are 0, 1, 4, 9, 3, 12, and 10, while the nonresidues are 2, 5, 6, 7, 8, and 11.

Quadratic Residues, II

When p is prime, there are $(p+1)/2$ quadratic residues modulo p:

Proposition (Quadratic Residues Mod p)

If p is prime, there are $(p+1)/2$ quadratic residues modulo p: specifically, the values 0^2, 1^2, 2^2, ... , $((p-1)/2)^2$.

Proof:

- All of these values are clearly quadratic residues.
- On the other hand, since $(p-x)^2 \equiv x^2 \pmod{p}$, there are no other possible squares modulo p, since the full list is simply $0^2, 1^2, \ldots, (p-1)^2$, and the second half of the list duplicates the first half.

If we consider only the nonzero residue classes mod p, this result says that exactly half of them are quadratic residues.

Quadratic Residues, III

We would like an easy way to detect quadratic residues, which we can do by computing the Legendre symbol:

Definition

If p is an odd prime, the Legendre symbol $\left(\frac{a}{p}\right)$ is defined to be 1 if a is a quadratic residue, -1 if a is a quadratic nonresidue, and 0 if $p|a$.

Examples:

1. We have $\left(\frac{2}{7}\right) = +1$, $\left(\frac{3}{7}\right) = -1$, and $\left(\frac{0}{7}\right) = 0$, since 2 is a quadratic residue and 3 is a nonresidue modulo 7.
2. We have $\left(\frac{3}{13}\right) = \left(\frac{-3}{13}\right) = +1$, and $\left(\frac{2}{15}\right) = 1$, since 3 and -3 are quadratic residues modulo 13, while 2 is not.

Quadratic Residues, IV

The notation for the Legendre symbol is somewhat unfortunate, since it is the same as that for a fraction inside parentheses; it is nonetheless standard.

- When appropriate, we may write $\left(\frac{a}{p}\right)_L$ to emphasize that we are referring to a Legendre symbol rather than a fraction.
- Note that the quadratic equation $x^2 \equiv a \pmod{p}$ has exactly $1 + \left(\frac{a}{p}\right)$ solutions modulo p.
- In general, if u is a primitive root modulo p, then a unit a is a quadratic residue if and only if it is an even power of u.
- Explicitly, if $a = u^{2k}$ then $(u^k)^2 = a$, and conversely if $a = b^2$ then $b = u^k$ is some power of u, and then $a = u^{2k}$ is an even power of u.

Quadratic Residues, V

Using this last observation we can give a much faster method for computing the Legendre symbol:

Theorem (Euler's Criterion)

If p is an odd prime, then for any residue class a, it is true that $\left(\frac{a}{p}\right) = a^{(p-1)/2} \pmod{p}$.

Example: Determine whether $a = 17441$ and $b = 135690$ are quadratic residues modulo the prime $p = 239441$.

- We simply compute $a^{(p-1)/2} \equiv a^{119720} \equiv 1 \pmod{p}$, so by Euler's criterion a is a quadratic residue mod p.
- Likewise, $b^{(p-1)/2} \equiv b^{119720} \equiv -1 \pmod{p}$, so by Euler's criterion b is not a quadratic residue mod p.

Quadratic Residues, VI

Proof:

- If $p|a$ then $\left(\frac{a}{p}\right) = 0 = a^{(p-1)/2} \pmod{p}$, so we win here.
- Now assume a is a unit modulo p and let u be a primitive root modulo p.
- If a is a quadratic residue, then $\left(\frac{a}{p}\right) = +1$.
- By the observation earlier, we know $a = u^{2k}$ for some integer k.
- Then $a^{(p-1)/2} \equiv (u^{2k})^{(p-1)/2} = (u^{p-1})^k \equiv 1^k = 1 \pmod{p}$, which agrees with $\left(\frac{a}{p}\right)$.

Quadratic Residues, VII

Proof (continued):

- Now suppose a is a quadratic nonresidue, so that $\left(\frac{a}{p}\right) = -1$.
- Then we must have $a = u^{2k+1}$ for some integer k, so $a^{(p-1)/2} \equiv (u^{2k+1})^{(p-1)/2} = (u^{p-1})^k \cdot u^{(p-1)/2} \equiv u^{(p-1)/2}$.
- Now observe that $x = u^{(p-1)/2}$ has the property that $x^2 \equiv 1 \pmod{p}$. The two solutions to this quadratic are $x \equiv \pm 1 \pmod{p}$, but $x \not\equiv 1 \pmod{p}$ since otherwise u would not be a primitive root as its order would only be $(p-1)/2$.
- Hence $u^{(p-1)/2} \equiv -1 \pmod{p}$, meaning that $a^{(p-1)/2} \equiv -1 \pmod{p}$ as well, and this agrees with $\left(\frac{a}{p}\right)$.

Quadratic Residues, VIII

As one of many corollaries of Euler's criterion, we can deduce that the Legendre symbol is multiplicative:

Corollary

For any odd prime p, the Legendre symbol modulo p is multiplicative: $\left(\frac{ab}{p}\right) = \left(\frac{a}{p}\right)\left(\frac{b}{p}\right)$. In particular, the product of two quadratic nonresidues is a quadratic residue.

Proof:

- $\left(\frac{ab}{p}\right) \equiv (ab)^{(p-1)/2} \equiv a^{(p-1)/2} b^{(p-1)/2} \equiv \left(\frac{a}{p}\right)\left(\frac{b}{p}\right) \pmod{p}$.

Quadratic Residues, IX

Finally, we will recall a useful result that allows us to compute square roots modulo a prime congruent to 3 modulo 4:

Proposition (Square Roots With $p \equiv 3 \pmod 4$)

If p is a prime congruent to 3 modulo 4 and a is a quadratic residue modulo p, then $x = a^{(p+1)/4}$ has $x^2 \equiv a \pmod p$.

Proof:

- By hypothesis we have $a = m^2 \pmod p$ by hypothesis and $m^{p-1} \equiv 1 \pmod p$ by Fermat's little theorem.
- Then $x^2 \equiv a^{(p+1)/2} \equiv m^{p+1} \equiv m^2 \equiv a \pmod p$, as claimed.

Quadratic Residues, X

We mention in passing that there are also ways to compute square roots mod p when $p \equiv 1 \pmod 4$ but they are a bit harder. Here is a general method for finding a root of a polynomial mod p:

Algorithm (Berlekamp's Root-Finding Algorithm)

Let $q(x) \in \mathbb{F}_p[x]$ and suppose that $q(x) = (x - r_1) \cdots (x - r_n)$ for some distinct $r_i \in \mathbb{F}_p$.

- *Choose a random $a \in \mathbb{F}_p$ and compute the gcd of $q(x - a)$ with $x^{(p-1)/2} - 1$ and $x^{(p-1)/2} + 1$ in $\mathbb{F}_p[x]$.*
- *If one of these gcds is a constant, choose a different value of a and start over.*
- *Otherwise, if both gcds have positive degree, then each gcd gives a nontrivial factor of $q(x)$.*
- *Repeat the factorization procedure on each gcd, until the full factorization of $q(x)$ is found.*

Quadratic Residues, XI

The idea behind Berlekamp's root-finding algorithm is that the quadratic residues mod p are essentially randomly distributed, and that they are the roots of $x^{(p-1)/2} - 1$.

- Thus, the greatest common divisor of $x^{(p-1)/2} - 1$ with $q(x)$ will be equal to the product of all the terms $x - r_i$ where r_i is a quadratic residue.
- So, if at least one root of q is a quadratic residue, and another is a quadratic nonresidue, then we will obtain a partial factorization of $q(x)$.
- The magic comes from working with $q(x - a)$, which shifts all of the roots of q by a. Since half of the residue classes modulo p are quadratic residues, we expect to obtain at least one quadratic residue and one quadratic nonresidue with probability roughly $1 - 2/2^n \geq 1/2$.

Quadratic Residues, XII

I will mention a few other nice connections between the Legendre symbol and group theory:

- First, the multiplicativity of the Legendre symbol tells us that it is a group homomorphism from the unit group $(\mathbb{Z}/p\mathbb{Z})^\times$ to the multiplicative group $\{\pm 1\}$.
- Euler's criterion tells us that this map can be computed explicitly as the $(p-1)/2$-power map.
- Since the Legendre symbol map is surjective, by the first isomorphism theorem, its kernel has index 2 (and that just a fancier way of saying that half of the unit residue classes are squares).

Quadratic Residues, XIII

Second, if a is a unit modulo p, then multiplication by a is a bijection on the units modulo p.

- Another way of saying this is: multiplication by a is a permutation in the symmetric group S_{p-1} on the $p-1$ unit residue classes.
- The Legendre symbol then assigns a value $+1$ or -1 to each of these permutations.
- From group theory, we have another way of assigning a value $+1$ or -1 to an arbitrary permutation: namely, to compute its sign (whether it is a product of an even or odd number of permutations).
- In fact, these two values agree with each other! This is a result known as Zolotarev's lemma (and you can optionally prove it on the homework).

Encoding Messages on Elliptic Curves, III

We can now return to the question of encoding messages on an elliptic curve $E : y^2 = x^3 + Ax + B$ modulo p, where we will now also take $p \equiv 3 \pmod 4$.

- Since half of the units modulo p are squares, for any given x there should exist a y with $y^2 = x^3 + Ax + B \bmod p$ about half of the time.
- If we try to encode a message directly as the x-coordinate of a point, we therefore should only expect to succeed about half of the time.
- A better procedure is instead to encode a message as part of the x-coordinate of a point, and then try to choose the remaining piece of the x-coordinate in such a way that $x^3 + Ax + B$ is a quadratic residue modulo p.

Encoding Messages on Elliptic Curves, IV

Here's one approach:

- Suppose p has $r + k + 1$ bits when written in base 2, we break the message into pieces each containing r bits.
- Then, to convert an r-bit message m, we pad the beginning m with $k + 1$ bits: a zero followed by k bits $b_1 b_2 \cdots b_k$ that can be arbitrarily chosen, and set x to be the bit string $0 b_1 \cdots b_k m$.
- Next, we search through the possible choices of these k bits until we find a solution y to $y^2 = x^3 + Ax + B \pmod{p}$, and pick one of the two possible values of y arbitrarily.
- We then perform our encryption procedure using the point (x, y) on E modulo p.
- To recover the message m from a point (x, y), where $0 \leq x < p$, we simply compute x modulo 2^r and write the result as a bit string in base 2.

Encoding Messages on Elliptic Curves, V

We can set the parameters in such a way that it is very likely we can find such a point for any given message piece r.

- Since there are 2^k possible choices for the bit string $b_1 b_2 \cdots b_k$, the probability that none of them yields a quadratic residue $x^3 + Ax + B$ is roughly $1 - 2^{-2^k}$.
- Of course, the probabilities are not entirely independent, but they should be fairly close to independent, certainly enough for a rough calculation like this.
- Even if we merely take $k = 10$, the failure probability is already so vanishingly small ($= 2^{-1024} \approx 1.8 \cdot 10^{-309}$) that it is unlikely a problem would ever occur in practical deployment.

Our calculation is also very efficient if we take $p \equiv 3 \pmod 4$, since then we can compute a square root of $x^3 + Ax + B$ using the proposition from earlier.

Encoding Messages on Elliptic Curves, VI

Example: Encode the message $m = 13 = 1101_2$ as a point on the elliptic curve $y^2 = x^3 + 11x + 17$ modulo $p = 307$ using a message length $r = 4$ bits and a padding length of $k = 4$ bits.

- We note that $p > 256 = 2^8$ so p has 9 bits in base 2.
- We therefore want to search for a bit string $b_1 b_2 b_3 b_4$ such that $x = 0 b_1 b_2 b_3 b_4 1101_2$ is a quadratic residue modulo 307.
- The bit string 0000 yields the value $x = 13$, but $x^3 + 11x + 17 \equiv 208 \pmod{307}$ is a quadratic nonresidue as can be confirmed by evaluating $208^{153} \equiv -1 \pmod{307}$.
- The bit string 0001, however, yields $x = 29$, and $x^3 + 11x + 1 \equiv 165 \pmod{307}$ is a quadratic residue as can be confirmed by evaluating $165^{153} \equiv 1 \pmod{307}$.

Encoding Messages on Elliptic Curves, VII

Example: Encode the message $m = 13 = 1101_2$ as a point on the elliptic curve $y^2 = x^3 + 11x + 17$ modulo $p = 307$ using a message length $r = 4$ bits and a padding length of $k = 4$ bits.

- To compute the associated value of y, we then compute $x^{(p+1)/4} \equiv 29^{77} \equiv 120 \pmod{307}$, since $p \equiv 3 \pmod{4}$.
- Thus, a point corresponding to the message m on the curve E is $(29, 120)$.
- To recover the message m, we simply extract the x-coordinate and reduce it modulo $2^4 = 16$. This yields the correct original message $13 = 1101_2$.

Encoding Messages on Elliptic Curves, VIII

Example: Encode the message $m = 13 = 1101_2$ as a point on the elliptic curve $y^2 = x^3 + 11x + 17$ modulo $p = 307$ using a message length $r = 4$ bits and a padding length of $k = 4$ bits.

- Of course, there are many other points on E that correspond to the same message m: another is the additive inverse $(29, 187)$ of the point we found.
- We could also have searched more randomly for possible bit strings (rather than starting at 0000 and going upward), to try to keep the procedure from being as predictable. The bit string 1110, for example, yields another possible point $(237, 209)$.

Elliptic Curve Cryptography

Now that we can convert messages into points on elliptic curves, we are ready to start creating public-key cryptosystems using elliptic curves.

- We will get into the details next time.
- But you might be surprised to learn that some cryptosystems designed for $\mathbb{Z}/m\mathbb{Z}$ do not really work at all for elliptic curves, while others will.
- We will also talk about how to use elliptic curves for key exchange and digital signature algorithms, since these are fairly closely related to public-key encryption.

Summary

We discussed elliptic curve factorization algorithms.

We discussed some cryptography fundamentals and reviewed some properties of quadratic residues.

We discussed how to encode messages as points on an elliptic curve.

Next lecture: Elliptic curve cryptography (part 2).