Lecture 29: Elliptic Curve Cryptography

William Stein

Math 124 Harvard University **Fall 2001**

Today's lecture is about an application of elliptic curves to cryptography.

Disclaimer: I do not endorse breaking laws, and give the examples below as a pedagogical tool in the hope of making the mathematics in our course more fun and relevant to everyday life. I don't think I have violated the Digital Millenium Copyright Act, because I have given very few details about Microsoft's actual protocols, and I've given absolutely no source code.

1 Microsoft Digital Rights Management

Today I will describe one way to use elliptic curves in cryptography. Our central example will involve version 2 of the Microsoft Digital Rights Management (MS-DRM) system, as applied to `.wma` audio files.

I learned about this protocol from a paper by "Beale Screamer".

1.1 Microsoft's Favorite Elliptic Curve

The elliptic curve used in MS-DRM is an elliptic curve over the finite field $k = \mathbb{Z}/p\mathbb{Z}$, where

$$p = 785963102379428822376694789446897396207498568951.$$

As Beale Screamer remarks, this modulus has high nerd appeal because in hexadecimal it is

$$89\text{ABCDEF}012345672718281831415926141424\text{F}7,$$

which includes counting in hexadecimal, and digits of e, π, and $\sqrt{2}$. The Microsoft elliptic curve E is

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

We have

$$\#E(k) = 785963102379428822376693024881714957612686157429,$$

and the group $E(k)$ is cyclic with generator

$$B = (771507216262649826170648268565579889907769254176,\\ 390157510246556628525279459266514995562533196655).$$

1.2 Nikita and Michael

Our heros Nikita and Michael love to share digital music  when they aren't out thwarting terrorists. When Nikita installed Microsoft's content rights management software on her compute, it sneakily generated a private key

$$n = 670805031139910513517527207693060456300217054473,$$

which it very stealthily hid in bits and pieces of files (e.g., `blackbox.dll`, `v2ks.bla`, and `IndivBox.key`). In order for Nikita to play Juno Reactor's latest hit `juno.wma`, her web browser contacts a Microsoft rights management partner. After Nikita gives Microsoft her credit card number, she is allowed to download a license to play `juno.wma`. Microsoft created the license using the ElGamal public-key cryptosystem (see below) in the group $E(k)$. Nikita's license file can now be used to unlock `juno.wma`, but *only* on Nikita's computer. When she shares both `juno.wma` and the license file with Michael, he is very annoyed because he can't play `juno.wma`. This is because Michael's computer doesn't know Nikita's computer's private key (that integer n above), so Michael's computer can't decrypt the license file.

`juno.wma`

2 The Elliptic Curve Discrete Logarithm Problem

Definition 2.1. If E is an elliptic curve over $\mathbb{Z}/p\mathbb{Z}$ and B is a point on E, then the *discrete log problem* on E to the base B is the following problem: given a point $P \in E$, find an integer n such that $nB = P$, if such an integer exists.

For example, let E be the elliptic curve given by $y^2 = x^3 + x + 1$ over the field $\mathbb{Z}/7\mathbb{Z}$. We have

$$E(\mathbb{Z}/7\mathbb{Z}) = \{\mathcal{O}, (2,2), (0,1), (0,6), (2,5)\}.$$

If $B = (2, 2)$ and $P = (0, 6)$, then $3B = P$, so $n = 3$ is a solution to the discrete logarithm problem.

To the best of my knowledge, the discrete logarithm problem on E is *really hard* unless $\#E(\mathbb{Z}/p\mathbb{Z})$ is "smooth", i.e., a product of small primes, or E is "supersingular" in the sense that $p \mid \#E(\mathbb{Z}/p\mathbb{Z})$. The Microsoft curve has neither of these deficiencies, and I expect that the discrete logarithm on that curve is quite difficult. This is not the weekness that "Beale Screamer" exploits in breaking MS-DRM.

3 ElGamal

How can we set up a public-key cryptosystem using an elliptic curve? The only public-key crytosystem that we've studied so far is the RSA cryptosystem; unfortunately, there is no analogue of RSA for elliptic curves! (Informal Exercise: Think about what goes wrong.)

MS-DRM uses the El Gamal system. Here's how it works. Start with a fixed, publicly known prime p, an elliptic curve E over $\mathbb{Z}/p\mathbb{Z}$, and a point $B \in E(\mathbb{Z}/p\mathbb{Z})$. Michael and Nikita choose random integers m and n, which are kept secret, and compute and publish mB and nB.

In order to send a message P to Michael, Nikita computes a random integer r and sends the pair of points $(rB, P + r(mB))$. To read the message, Michael multiplies rB by his secret key m to get $m(rB) = r(mB)$, and subtracts this from the second point to get

$$P = P + r(mB) - r(mB).$$

As far as I can tell, breaking this cryptosystem requires solving the discrete logarithm problem, so it's very difficult.

The following example is based on an example taken from Beale Screamer's paper.

Example 3.1. Nikita's license files contains the pair of points $(rB, P+r(nB))$, where

$$rB = (179671003218315746385026655733086044982194424660, 697834385359686368249301282675141830935176314718)$$

and

$$P + r(nB) = (1378510385482644673726451580930040003436391189 15, 1108485892286762240572292302235808150242248756 99).$$

Nikita's computer sneakily loads the secret key

$$n = 670805031139910513517527207693060456300217054473$$

into memory and computes

$$n(rB) = r(nB) = (328901393518732637577115650601768681044040715701, 586947838087815993601350565488788846203887988162).$$

It then subtracts this from $P + r(nB)$ to get

$$P = (14489646124220757767, 669337780373284096274895136618194604469696830074).$$

That x coordinate, 14489646124220757767, is the top secret magic "content key" that unlocks `juno.wma`.

If Nikita knew the private key n that her computer generated, she could compute P herself and unlock `juno.wma` and share her music with Michael, just like she used to share her favorite CDs with Michael. Beale Screamer found a weakness in Microsoft's system that let him find n:

> "These secret keys are stored in linked lists ... interspersed with the code in the library. The idea is that they can be read by that library, used internally by that library, and never communicated outside the library. Since the `IndivBox.key` file is shuffled in a random way for each client, these keys would be extremely difficult to extract from the file itself. Fortunately, we don't have to: these keys are part of the object state that is maintained by this library, and since the offset within this object of these secret keys is known, we can let the library itself extract the secret keys! The code for this simply loads up the 'black box' library, has it initialize an instance of the object, and then reads the keys right out of that object. This is clearly a weakness in the code which can be corrected by the DRM software fairly easily, but for now it is the basis of our exploit."

As you can see, Microsoft has undertaken a difficult and interesting problem. *How can Microsoft store data on Nikita's computer in such a way that Nikita can not access it, but Nikita's computer can?*

4 Why Use Elliptic Curves?

There are several advantages to using elliptic curves instead of $\mathbb{Z}/p\mathbb{Z}$ for cryptography, though the people at RSA Corporation might disagree. Elliptic curve cryptosystems with smaller key sizes appear to be just as secure as "classical" $\mathbb{Z}/p\mathbb{Z}$ cryptosystems with much larger key sizes, so elliptic curve cryptosystems can be more efficient. Another advantage, which I won't explain at all, is that elliptic curve cryptosystems appear to be vastly more secure over "large finite fields of characteristic 2" than RSA, which is is very important in practical applications. Also, elliptic curves are simply way cooler than $\mathbb{Z}/p\mathbb{Z}$, so they (used to) attract venture capitalists.

Some mobile phones also use elliptic curve cryptography. Do you have an elliptic curve in your pocket right now?