Elliptic Curves In Cryptography

by Atticus Kuhn
December 2019

CONTENTS

INTRODUCTION

I will look at the fascinating subject of elliptic curves as they pertain to cryptography, all the way from their past to possible usage in the future.These extraordinary curves provide the underpinnings for many modern cryptography protocols

HISTORY

Elliptic curves were somewhat studied by Diophantus in the 3rd century, but real work began in the 18^{th} century when mathematicians were trying to find the arc length of an ellipse. It was not until 1985 that Koblitz and Miller realized their potential for cryptography.

STRUCTURE OF ELLIPTIC CURVES

Elliptic Curves satisfy the equation $y^2 = x^3 + ax + b$, where a and b are constants such that $4a^3+27b^2 \neq 0$. There are many ways of expressing an elliptic curve, but this is called Weierstrass form.

Remark: This condition, $4a^3 + 27b^2 \neq 0$, is because the cubic formula says a cubic $y = ax^3 + bx^2 + cx + d$ has the discriminant $b^2c^2 - 4ac^3 - 4b^3d - 27a^2d^2 + 18abcd$. If we plug in $(a, b, c, d) = (1, 0, a, b)$, we get $4a^3 + 27b^2$. Since we do not want double roots, the discriminat cannot be 0.

Elliptic Curves as a group: Elliptic curves can be defined over many sets, but we will look at the reals and integers mod p, where p is a prime. A nice property of Elliptic curves is that they form an abelian group. Let us look at the properties of a group, namely a binary operation, and identity as they pertain to Elliptic Curves. A group requires an operation which is called addition. If the curve is defined over the reals, we can add points graphically. To add 2 points, P and

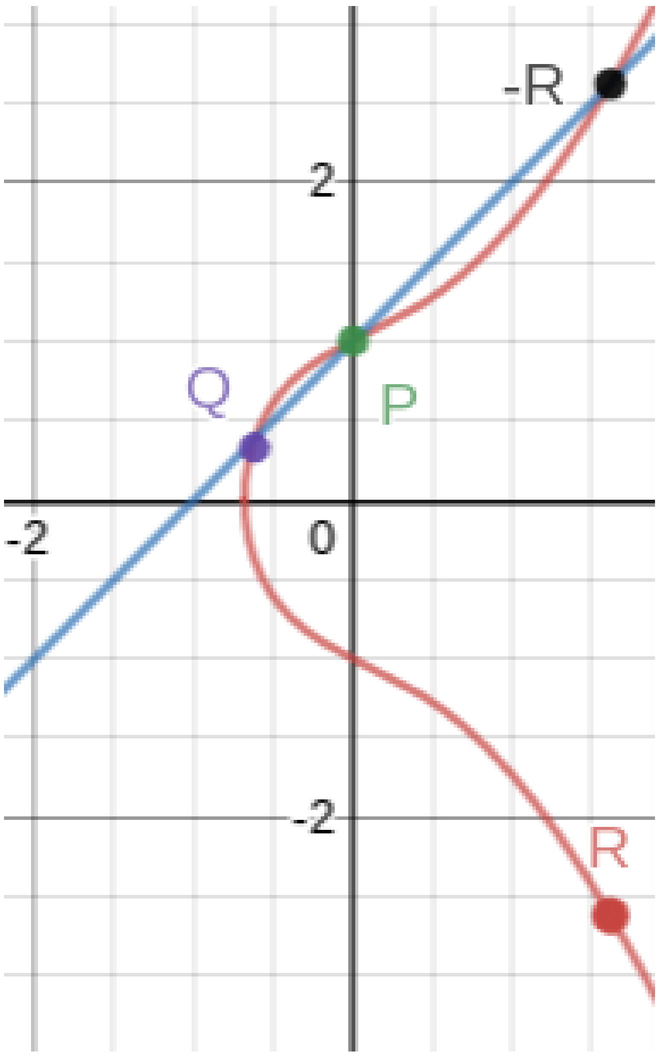

Figure 1. Addition over $\mathbb{R}$ in Elliptic Curves

Q, extend the line until it hits the curve in a 3rd point, -R, and then reflect that over the x-axis to get $R := P + Q$.

We can add a point to itself, by drawing the tangent line to the curve at the point. By implicit differentiation, the slope of the tangent line is $\frac{dy}{dx}2y = 3x^2 + a$, or$\frac{dy}{dx} = \frac{3x^2+a}{2y}$

Remark: If a line intersects 2 non-vertical points on a elliptic curve, then why will it always intersect at a 3rd point? We will see why it must be true when we explore algebraically adding points.

The identity of an elliptic curve is the point at infinity, sometimes written as $\mathcal{O}$. The addition of any point on the curve to $\mathcal{O}$ is itself, and the addition of 2 points with the same x-coordinate is $\mathcal{O}$.

Before, we explored a graphical method of adding 2 points, but it can sometimes be useful to have an algebraic method as well. If we are looking at $\mathbb{F}_p$, then it makes more sense to use an algebraic approach. Let $P = (x_p, y_p)$, and $Q = (x_q, y_q)$, be points on the curve $y^2 = x^3 + ax + b$. Both P and Q lie on the line $y - y_p = (\frac{y_p - y_q}{x_p - x_q})(x - x_p)$ or $y = (\frac{y_p - y_q}{x_p - x_q})x - (\frac{y_p - y_q}{x_p - x_q})x_p + y_p$ For simplicity, let $m = (\frac{y_p - y_q}{x_p - x_q})$ We can now substitute in y to the equation $y^2 = x^3 + ax + b$ to get $(mx - mx_p + y_p)^2 = x^3 + ax + b$. Subtracting, we get $0 = x^3 + ax + b - (mx - mx_p + y_p)^2$ or $0 = x^3 + m^2x^2 + (2x_qm + 2x_pm^2 + x_p)x + b + y_p^2m^2 - 2x_px_qm + x_q^2$. Because both P an Q make the equation true, by the factor theorem, $(x - x_p)(x - x_q)$ must divide the equation. Since a cubic has 3 solutions, the 3rd point is the other solution to the equation. Vieta's theorem tells us that the sum of the roots will be m^2, so the 3rd root must be $m^2 - x_p - x_q$. That is the x coordinate, and the y coordinate is achieved by plugging it into the line $y = mx - mx_p + y_p$ to get $m^3 - 2mx_p - mx_q + y_p$.

If we want to add a point to itself algebraically, we just use the value of m we calculated before as $\frac{3x_p^2 + a}{2y_p}$

We can also multiply a point by an integer, which is achieved by repeatedly adding a point to itself.

COMMUNICATION WITH ELLIPTIC CURVES

Communication with Elliptic Curves mainly focuses on using $\mathbb{Z}/p\mathbb{Z}$, but before we can communicate with elliptic curves, we must first look at a point P. Each point makes a cyclic group, where all points in the group can be achieved through some multiplication of P. The order of P is the number of points in the cyclic group, and is written ord(P). To measure how big the cyclic group of P is, we use the co factor. The co factor is written as h, and is the number of points on the Elliptic Curve in $\mathbb{Z}/p\mathbb{Z}$ divided by the order of P. It is expressed as $h = \frac{|E(\mathbb{Z}/p\mathbb{Z})|}{ord(P)}$.

One Protocol for communicating with Elliptic Curves is Diffie-Hellman/El Gamal, which can be implemented in all groups, but is only practical in groups that are hard to crack. The Elliptic Curve version is similar to the discrete logarithm version, except powers mod p are replaced with Elliptic multiplication mod p. Let Alice want to send a message m to Bob over an insecure channel without Eve knowing the message. Alice and Bob can publicly discuss the parameters of the encryption, such as what prime they will use and what curve $y^2 = x^3 + ax + b$ they will use. They can also publicly reveal what point P they will use, as well as the order and co factor of P. Both Alice and Bob privately choose a number, α and β respectively, both of which are within the order of P. They both publish their private key times P, so $A = \alpha * G$ or $B = \beta * G$. Bob then multiplies A by his secret key to get $A\beta = \alpha\beta * G$, and Alice multiplies B by her secret key to get $B\alpha = \alpha\beta * G$. Now the key exchange is complete, Bob and Alice both have the same key, so they can now communicate.

Alice encrypts her message m by adding it by the key to get c = m $+\alpha\beta * G$. She then publishes c. Bob recovers the message by subtracting the key to get $c - \alpha\beta * G = m + \alpha\beta * G - \alpha\beta * P = m$.

	Alice	Eve	Bob
Agree on Domain Parameters	p, a,b, G, ord(G)	p,a,b, G, ord(G)	p,a,b , G, ord(G)
Pick Private Key	α		β
Publish keys	p, $\alpha\beta * G$	A, B	$\alpha\beta * G$
Send Message	c, m	c	c, m

CRACKING ELLIPTIC CURVES

The question arises from the encryption protocol: why can't Eve figure out $\alpha\beta * G$ from A and B? This is the Elliptic Curve Discrete Logarithm Problem (ECDLP). ECDLP says that if Q is a point on an elliptic curve mod p, such that Q is a multiple of a point P, then it is hard to find n such that Pn= Q.

Although Elliptic Curve Diffie-Hellman is similar to regular Diffie Hellman, it is not susceptible to specialized attacks, such as the Index Calculus. This means that for a message to be equally secure (for Eve to take the same amount of time to crack it) in both ElGamal, and Elliptic, it requires a smaller key for Elliptic. This is measured in security level, meaning if a protocol and key has n-bit security, then Eve needs 2^n computations to crack it.

In the future, it is possible that elliptic curves could be broken by Shor's algorithm in polynomial time (or at least subexponential time). Peter Shor invented 2 algorithms that could run a quantum computer to easily factor numbers or compute the discrete logarithm problem. The version which solves the Discrete Logarithm Problem (DLP) can be adapted to solve ECDLP. A fascinating paper by Martin Roetteler, Michael Naehrig, Krysta M. Svore, and Kristin Lauter (2017) found a way to do break ECC iwth about 1000 qubits. They even suggest that ECC is more vulnerable to quantum computers then RSA. Luckily, Supersingular isogeny Diffie Hellman key exchange (SIDH) was invented and seems to be resistant to quantum computers. An isogeny is a morphism, or special type of mapping. ϕ: $E_1 \rightarrow E_2$ that maps one elliptic curve to another. The main principle is that given $E, \phi(E)$, it is hard for even quantum computers to find ϕ. Explaining the details of SIDH is beyond the scope of this paper.

OTHER USES OF ELLIPTIC CURVES

Elliptic Curves have many other uses other than cryptography, namely digital signatures.

Digital signatures solve the problem of identity verification. Perhaps Eve pretends to be Bob. Then Alice sets up encrypted communication with Eve, and Eve can easily decrypt the messages.

A nice variation of digital signatures utilizes elliptic curves and is called Elliptic Curve Digital Signature Algorithm (ECDSA). Say Alice wants to sign a message m. We will assume Alice and Bob have already agreed on the parameters of the encryption (p, P, etc), and Alice has already made her private key (α) public key (A) pair. She first computes the hash of m.

Remark: Alice and Bob can agree on a hash function in advance, as long as it is a deterministic one-way function that always returns an answer of fixed length. and is hard to reverse.

Alice then lets z be the ord(P) leftmost bits of the hash. Alice must then choose a random number k in ord(P). She finds the point on the curve which is $k * G$ and takes the x coordinate, call it r. Alice finds the inverse of k mod p so she can write s = $k^{-1}(z + r * \alpha)(\mod p)$. Alice then sends Bob the r and s. For Bob to check the signature, he first computes z in the same way that Alice did. Bob computes the inverse of s, and finds the point $zs^{-1} * G + rs^{-1} * A$. He then checks if the x value of the point is r.

Why does Bob get back r if the signature is valid? Let us re-write the expression as $zs^{-1} * P + rs^{-1} * A = zs^{-1} * P + rs^{-1} * \alpha * P$. Multiplication of elliptic curves is distributive, so $zs^{-1} * P + rs^{-1} * \alpha * P = (z + r\alpha)s^{-1}P$. But remember that s was defined as $k^{-1}(z + r * \alpha)$, so we can substitute $(z + r\alpha)s^{-1}P = (z + r\alpha)(k^{-1}(z + r * \alpha))^{-1}P = (z + r\alpha)k(z + r * \alpha)^{-1}P = kP$. Recall that r was defined as kP, so we Bob has calculated r.

Additionally, only Alice could have produced this signature, because she is the only one who knows α.

CAUTIONARY TALES OF ELLIPTIC CURVES

Just because elliptic curves can be very secure and fast, that does not mean they are impervious. One still needs to follow all proper precautions when using elliptic curves. Perhaps the most famous failure of elliptic curves is the Dual Elliptic Curve Deterministic Random Bit Generator, abbreviated as Dual EC DRBG. An important aspect of cryptography is picking random numbers: Alice and Bob both had to pick random private keys in El Gamal and Alice had to pick a random k for digital signatures. With primes unimaginable large, and an intense volume of encrypted messages sent each day, it seems necessary that for an algorithm to pick random numbers for us. Thus the National Institute for Standards and Technology, in 2006, set Dual EC DRBG as a national standard. The way a normal Psuedo-Random Number generator works is it starts off with a seed, s_0, and uses 1 trapdoor function to get the secret internal state s_1. It uses a different trapdoor function on s_1 to get r_1.This process is repeated. A key property is that if a PRNG makes a string of numbers, a malicious actor should not be able to predict future outputs from initial outputs. In Dual EC DRBG, the trapdoor functions were multiplication by points on an elliptic curve, P and Q. So, for example, $s_1 = P * s_0$, and $r_1 = Q * s_1$, and the first 16 digits of the output were deleted. The monumental failure of Dual EC DRBG was that P and Q were chosen very specific way. Usually, if variables are hard coded into cryptography equations, they will be Nothing Up My Sleeve numbers (NUMS), such as the first digits of pi to prove that the inventor did not specifically choose the numbers to have a special property. It is highly likely that the Nation Security Administration (NSA), chose P and Q such that there was a number d so $P * d = Q$. That means if the NSA knew $r_1 = s_1 * P$, then they could multiply r_1 by d to get $r_1 = s_1 * P * d = s_1 * Q = s_2$. Now that the NSA knows the internal state of the PRNG, it can predict all future outputs.

MODERN USES OF ELLIPTIC CURVES

To prevent man-in-the-middle attacks, where Eve relays messages between a website and a client by pretending to be a website, https uses a digital certificate. A website gets a certificate by having a well known and trusted certificate authority sign its public key. Now Eve cannot impersonate the website unless she gets the certificate authority to sign her public key as well.

An example of this can be found if you go to https://en.bitcoin.it/wiki/. Click the padlock to the left of the URL, and click on Certificate(valid). Then go to the details tab, and under Certificate Signature Algorithm, you will see X9.62 ECDSA Signature with SHA-256. This means the certificate was made using elliptic curves.

In fact, Bitcoin itself uses ECDSA, specifically using the parameters defined by Secp256k1.

KEYWORDS

α: Alice's public key;

β: Bob's public key;

A: Alice's private key;

B: Bob's private key;

a,b: Coefficients of the elliptic curve $y^2 = x^3 + ax + b$;

Alice, Bob: fictional characters who want to communicate securely over a public channel;

Eve: an eavesdropper who wants to decrypt the messages of Alice and Bob;

p: an arbitrary prime;

m: message that Alice wants to send Bob;

Weierstrass form: Typical form of an elliptic curve, $y^2 = x^3 + ax + b$;

P,Q: arbitrary points on an elliptic curve;

r: outputs of a PRNG;

s: the internal state of a PRNG;

PRNG: a pseudo random number generator;

Hash Function: a function with 4 key properties. It is quick and deterministic to output a fixed length hash of any input, it is hard to reverse engineer, it is hard to find 2 inputs with the same hash, and a small change in the input leads to large changes in the hash;

SOURCES

Classroom, Sunny. “Why Digital Certificate?” YouTube, YouTube, 2018, `www.youtube.com/watch?v=UbM1PIgzTxc`. Costello, Craig, et al. “Efficient Algorithms for Supersingular Isogeny Diffie-Hellman.” Advances in Cryptology – CRYPTO 2016 Lecture Notes in Computer Science, 2016, pp. 572–601., doi:10.1007/978-3-662-53018-4_21.

“Elliptic Curve Digital Signature Algorithm.” Elliptic Curve Digital Signature Algorithm - Bitcoin Wiki, `en.bitcoin.it/wiki/Elliptic_Curve_Digital_Signature_Algorithm`.

“Elliptic Curve Digital Signature Algorithm.” Wikipedia, Wikimedia Foundation, 29 Oct. 2019, `en.wikipedia.org/wiki/Elliptic_Curve_Digital_Signature_Algorithm`.

“Elliptic-Curve Cryptography.” Wikipedia, Wikimedia Foundation, 15 Nov. 2019, `en.wikipedia.org/wiki/Elliptic-curve_cryptography`.

`eprint.iacr.org/2017/598.pdf`.

Jao, David, and Luca De Feo. “Towards Quantum-Resistant Cryptosystems from Supersingular Elliptic Curve Isogenies.” Post-Quantum Cryptography Lecture Notes in Computer Science, 2011, pp. 19–34., doi:10.1007/978-3-642-25405-5_2.

Koblitz, Neal. “Elliptic Curve Cryptosystems.” Mathematics of Computation, vol. 48, no. 177, 1987, p. 203., doi:10.2307/2007884.

Martin, et al. “Quantum Resource Estimates for Computing Elliptic Curve Discrete Logarithms.” ArXiv.org, 31 Oct. 2017, `arxiv.org/abs/1706.06752`.

“Nothing-up-My-Sleeve Number.” Wikipedia, Wikimedia Foundation, 5 Nov. 2019, `en.wikipedia.org/wiki/Nothing-up-my-sleeve_number`.

quanta, quanta. “What Is Isogeny?” Mathematics Stack Exchange, 1 July 2018, `math.stackexchange.com/questions/36724/what-is-isogeny#36728`.

Ravanel, Doug. “Elliptic Curves: What They Are, Why They Are Called Elliptic, and Why Topologists like Them.” Rochester, 2017, `web.math.rochester.edu/people/faculty/doug/mypapers/wayne1.pdf`.

Roetteler, Martin, et al. “Quantum Resource Estimates for Computing Elliptic Curve Discrete Logarithms.” Advances in Cryptology – ASIACRYPT 2017 Lecture Notes in Computer Science, 31 Oct. 2017, pp. 241–270., doi:10.1007/978-3-319-70697-9_9.

“Secp256k1.” Secp256k1 - Bitcoin Wiki, en.bitcoin.it/wiki/Secp256k1.

“Security Level.” Wikipedia, Wikimedia Foundation, 10 Feb. 2019, `en.wikipedia.org/wiki/Security_level`.

Sosa, Pedro. “Introduction to Supersingular Isogeny Difffie-Hellamn.” Academia.edu, 2017, `www.academia.edu`.

“Supersingular Isogeny Key Exchange.” Wikipedia, Wikimedia Foundation, 11 Nov. 2019, `en.wikipedia.org/wiki/Supersingular_isogeny_key_exchange`.

Rubinstein-Saledo, Simon (2019). *Cryptography. Springer.*

Silverman, J. & Tate, J. (2015). *Rational Points on Elliptic Curves. Second Edition. Springer.*

Washington, L. (2008). *Elliptic Curves Number Theory and Cryptography. Second Edition. Chapman & Hall.*