Elliptic Curve Cryptography

Christine Hjelmfelt
advisor Dr. Patrick Fleming

December 2015

Table of Contents

Introduction

Cryptocurrencies such as Bitcoin use cryptography to secure transactions. Cryptocurrencies are a type of digital coin. They are not created by a government but instead they are developed by programmers, distributed by algorithms and secured by mathematical principles. There are currently thousands of cryptocurrencies in the world. Bitcoin is currently the largest cryptocurrency in the world with a market value of over 3.5 billion dollars. Understanding what cryptocurrency is and how it works is mostly a computer science problem, however at the heart of it lies this question: "How do we know who owns a digital coin?"

The mathematics is in the process of authenticating a transaction. In order to spend coins you must generate a signature that proves you are the owner. Elliptic curve cryptography is used to create these digital signatures. The algorithms prevent signatures from being forged, therefore no one can claim to own someone else's coins. Elliptic curve cryptography is used in many other applications such as sending private messages, server authentication, and the encryption of digital documents.

To understand how elliptic curve cryptography is used in cryptocurrency we will need to explore the following concepts: We need to understand fields, field extensions and basic cryptosystems. We look at a simple example of a cryptosystem to see how fields are used to comunicate securely. Then we will look at the basic concept of eliptic curves and define an addition on an elliptic curve. We need to explore how an eliptic curve may be used to define a group. It is necessary to find points on the curve. We can add two points on a curve to get a third point, this will be necessary for the calculations in our cryptosystem. We will consider how elliptic curves can be used to pass secure messages. Then finally we look at how elliptic curves are used to create a signature that proves ownership of a coin.

Fields

A field is a set of elements with two operations, addition and multiplication. It is closed under both addition and multiplication, there is an additive identity and a multiplicative identity, there are inverses for all elements under addition and inverses for all elements except for 0 under multiplication, both addition and multiplication are associative and commutative, and multiplication distributes over addition. Examples of fields include: real numbers, complex numbers, and rational numbers.

Finite Fields

A finite field $\mathbb{F}_P$ is a set with a finite number of elements. It is of prime order or a power of a prime. It has two binary operations, addition and multiplication. Addition forms a cyclic group and multiplication forms an abelian group (excluding 0).

Properties

It is closed under both addition and multiplication
There is an additive identity, 0
There is a multiplicative identity, 1
There are inverses for all elements under addition, $a + a^{-1} = 0$
There are inverses for all elements (except 0) under multiplication, $aa^{-1} = 1$
Addition is associative, (a + b) + c = a + (b + c)
Multiplication is associative, (ab) c = a (bc)
Addition is commutative, a + b = b + a
Multiplication is commutative, ab = ba
Multiplication distributes over addition, a (b + c) = ab + ac

Example
An example of a prime order field is $\mathbb{F}_5 = \{0, 1, 2, 3, 4\}$
All operations are done mod(5)

$3 + 4 = 7 \bmod(5) = 2$
$3 * 2 = 6 \bmod(5) = 1$

+	0	1	2	3	4
0	0	1	2	3	4
1	1	2	3	4	0
2	2	3	4	0	1
3	3	4	0	1	2
4	4	0	1	2	3

*	0	1	2	3	4
0	0	0	0	0	0
1	0	1	2	3	4
2	0	2	4	1	3
3	0	3	1	4	2
4	0	4	3	2	1

Extending a Field of Prime Order
We can use a field of prime order to define a field of non-prime order by extending the prime order field with field extensions. We take $\mathbb{F}_P$ and look for a polynomial of degree n that is irreducible over $\mathbb{F}_P$. To extend $\mathbb{F}_P$ to $\mathbb{F}_{P^k}$ we find a polynomial $P(x)$ of degree k with no roots in $\mathbb{F}_P$. Let ω be a root of $P(x)$. $P(\omega) = 0$ then solve for ω^k. This polynomial allows us to define the set of elements in our new field $\mathbb{F}_{P^k}$.

Example
$\mathbb{F}_9 = \mathbb{F}_{3^3}$ where $\mathbb{F}_3 = \{0, 1, 2\}$
We need a 3rd degree polynomial with no roots in $\mathbb{F}_9$.
$\omega^3 + 1 = 0$ has no roots.
If we then make ω^3 a root. $\omega^3 = -1 \bmod(3) = 2$
Then we can find the elements of $\mathbb{F}_9$
$\mathbb{F}_9 = \{0, 1, 2, \omega, \omega + 1, \omega + 2, \omega^2, \omega^2 + 1, \omega^2 + 2\}$

Fields of Characteristic 2

A field of characteristic 2 is an extension from $\mathbb{F}_2$ to a higher order $\mathbb{F}_{2^n}$. Fields of characteristic 2 are a bit different to work with. They are useful because they allow us to use binary calculations. Computers use binary operations and we can take advantage of those operations to speed up the calculations. Since a secure crypto system requires enormous numbers of calculations this difference will be substantial.

Example
We start with $\mathbb{F}_2$:
This gives us a set of points: $\mathbb{F}_2 = \{0, 1\}$

$1 + 1 = 2 \bmod(2) = 0$
$1 * 1 = 1 \bmod(2) = 1$

+	0	1
0	0	1
1	1	0

*	0	1
0	0	0
1	0	1

Extending $\mathbb{F}_2$ to higher order
We need a huge number of points for a secure crypto system. It is necessary to extend $\mathbb{F}_2$ to a much larger field. To extend $\mathbb{F}_2$ to $\mathbb{F}_{2^n}$ we need a polynomial of degree n that is irreducible over $\mathbb{F}_2$.

To extend to $\mathbb{F}_8$ we find $P(x) = x^3 + x + 1$. It cannot be factored.
Let ω be a root of P(x), then $P(\omega) = 0$, so $\omega^3 = \omega + 1$
This polynomial allows us to define our field:
$\mathbb{F}_8 = \{0, 1, \omega, \omega + 1, \omega^2, \omega^2 + 1, \omega^2 + \omega, \omega^2 + \omega + 1\}$

Operations are done mod(2), where $\omega^3 = \omega + 1$.

$\omega + 1 + \omega^2 + \omega + 1 = 2\omega + 2 + \omega^2 \bmod(2) = 0 + 0 + \omega^2 = \omega^2$
$\omega(\omega^2 + 1) = \omega^3 + \omega \bmod(2) = \omega + 1 + \omega \bmod(2) = 2\omega + 1 \bmod(2) = 1$

Discrete Log Problem

Since we have multiplication of elements then we also have integer exponents.
$a * a * a * a * a = a^5$
This also means we have logs. If we have $a^k = b$, finding b is easy if you have a and k, however, finding k is hard even if you have a and b. A computer can calculate a^k extremely fast even when both a and k are very large numbers. For large numbers, finding k given a and b takes a tremendously long time even on a super computer. We can choose numbers that are so big even the most advanced computers we can forsee being built will take millions of years to calculate the value of k while an ordinary computer if given a and k can find b very quickly. This is called the discete log problem and it is used as the basis for a number of cryptosystems.

Example
Using $\mathbb{F}_{23}$:
Exponent: $17^3 = x \bmod(23)$
This is easy to solve: $17^3 = 4913 \bmod(23) = 14 \bmod(23)$
Log: $17^b = 14 \bmod(23)$
This is hard to solve. It requires special techniques and a lot of time.

Public Key Cryptosystem (Discrete Log)

Alice and Bob want to send private messages to one another. By using a cryptosystem they can be assured of three things: the message can only be read by the person they send it to, the sender cannot claim someone else sent it, and someone else cannot claim they were the sender.

To encrypt and decrypt messages they need these things:
$\mathbb{F}_P$, a prime number p gives us a set $\mathbb{F}_P = 0, 1, 2, ...p - 1$
G, a generator of $\mathbb{F}_P$ (an integer that when multiplied by itself repeatedly gives all of $\mathbb{F}_P$ except for 0)
m, the message (an integer in $\mathbb{F}_P$)
a, a private key for Alice
$\mathbf{A} = G^a$, a public key for Alice
b, a private key for Bob
$\mathbf{B} = G^b$, a public key for Bob
M, the encrypted message

Alice and Bob agree on which prime number to use and the choice of generator. They create their public keys by taking the generator to the power of their private key. They then exchange these public keys.
Now Bob can send Alice a message, m. He encodes his message with her public key and his private key:
$M = m * A^b = m * G^{ab}$
Bob sends the encoded message M to Alice.

She decodes the message using his public key and her private key:
$m = M * B^{-a} = M * G^{-ab} * G^{ab}$
Now she can read the original message m.

The public keys A and B, might be available for anyone to look up but in order for a person to decrypt a message they need either one of the private keys, a or b.

Example
Bob and Alice want to send messages using the prime number 13. This gives the set $\mathbb{F}_P = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12$. They find that 11 is a

generator of $\mathbb{F}_P$ and calculate public keys. Alice chooses private key $a = 2$ and generates her public key $A = G^a = 11^2 = 121 \mod(13) = 4$. Bob chooses private key $b = 3$ and generates his public key $B = G^b = 11^3 = 1331 \mod(13) = 5$. They then exchange the public keys.

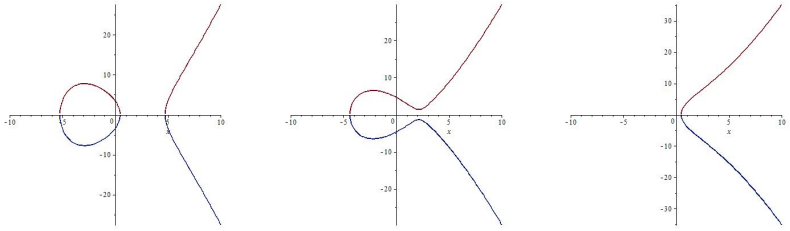
Bob has a message he wants to send. Each possible message corresponds to an integer in P. His message is $m = 12$. He encrypts m with her public key and his private key:
$M = m * A^b = 12 * 4^3 = 768 \mod(13) = 1$
Bob sends the encrypted message M to Alice.
Alice recieves the message and to decrypt it she uses his public key and her private key:
$m = M * B^{-a} = 1 * 5^2 = 25 \mod(13) = 12$
She can now read the unencypted message m.

Elliptic Curve Cryptosystems

We now delve into eliptic curves and the cryptosystems made possible by them. This will cover the basics of elliptic curves, elliptic curve groups, adding points on the curve, elliptic curve public key cryptograph, and finally digital signatures.

Elliptic Curves

An elliptic curve is the set of points that satisfy the equation:
$y^2 = x^3 + ax + b$
or for fields of characteristic 2:
$y^2 + xy = x^3 + ax^2 + b$
The choice of a and b gives the particular shape of the curve.



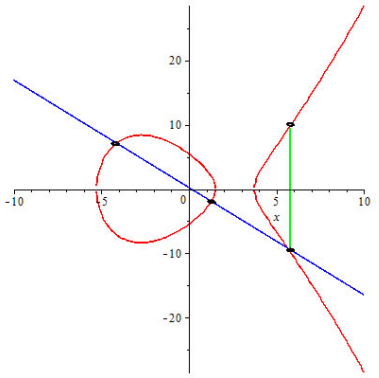
These are eliptic curves over the set of all real numbers.

Elliptic Curve Groups

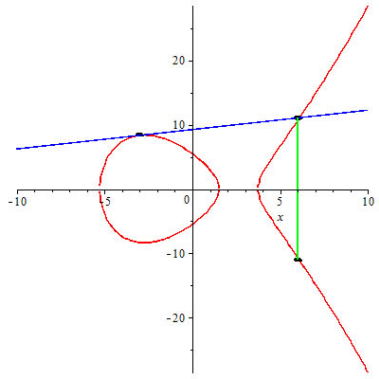
The set of points on the curve plus an additional point at infinity, O, which is called "zero", will form an elliptic curve group under addition, which I will define shortly. In order to prove the properties of elliptic curves we first need to know how to add points on the curve. First I will give some explaination and examples, then I will prove it is a group.

Defining Addition of Points
We can add two points on the elliptic curve together

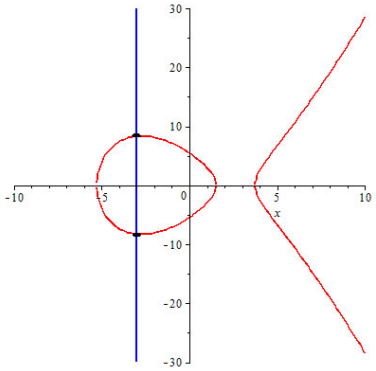
Case 1: Distinct x values
A point P and a point Q form a line that crosses at a third point we call -R. We then negate it to find point R.



Case 2: Doubling
If we add point P to itself we use the tangent line to find the third point. This is called doubling.



Case 3: Same x values
If the two points P and Q have the same x value the line intersects at the third point, O. So P and Q are additive inverses.



Finding Points
Example: $\mathbb{F}_{23}$ We want a curve that gives us a prime number of points
The elliptic curve equation is $y^2 = x^3 + ax + b$. a and b can be any integer in $[0, 22]$. If we choose $a = 5$ and $b = 3$ we get Eliptic Curve: $y^2 = x^3 + 5x + 3$.

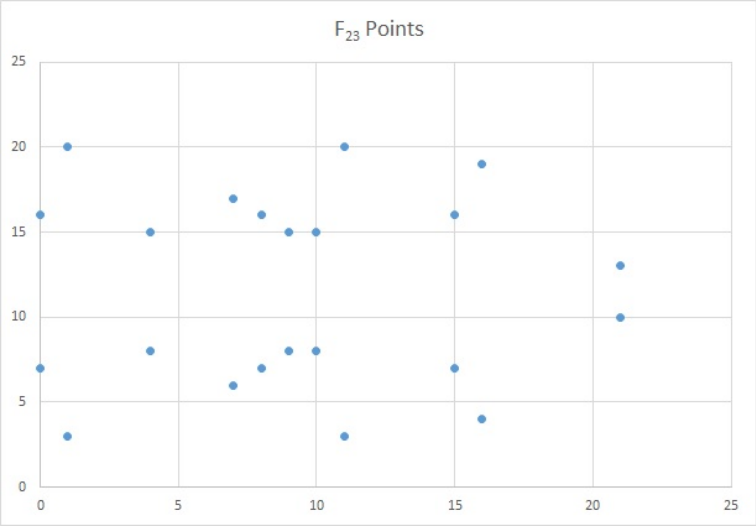
We plug in 0-22 for x. Only some values of x will give us a y^2 value. For those cases there should be two y values. If there are not two it is a degenerate case and we don't use that curve. Adding the O point should give us a prime number of points.

when $x = 0$: $\quad y^2 = 0^3 + 5(0) + 3 \quad y^2 = 3$
$y = 7 \quad 7^2 = 49 \bmod 23 = 3$
$y = 16 \quad 16^2 = 256 \bmod 23 = 3$

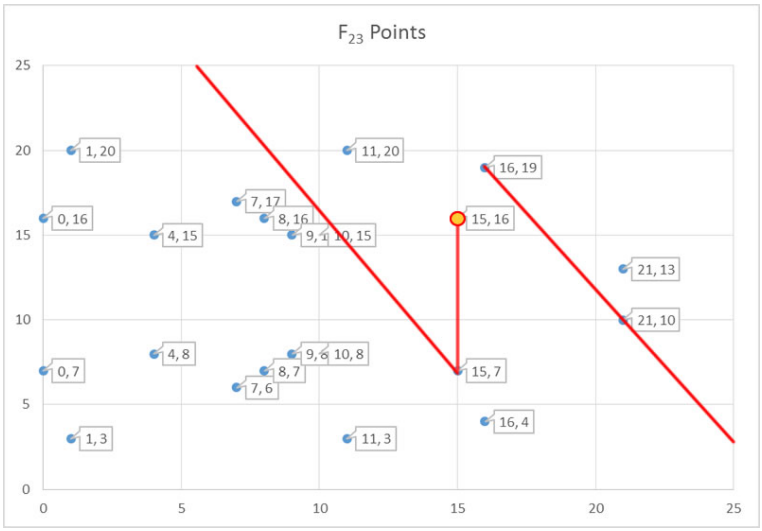
Points on the Curve in $\mathbb{F}_{23}$
Once we have all of our x,y values they give us the points on the curve:

(0,7)
(0,16)
(1,3)
(1,20)
(4,8)
(4,15)
(7,6)
(7,17)
(8,7)
(8,16)
(9,8)
(9,15)
(10,8)
(10,15)
(11,3)
(11,20)
(15,7)
(15,16)
(16,4)
(16,19)
(21,10)
(21,13)
O

This is the graph of the points on our curve



The curve is difficult to recognize because the points are mod(23). But we can still add points just as we did over the real numbers.



$(16, 19) + (21, 10) = (15, 16)$

Equations

We need a set of formulas to calculate the lines and the points they cross. If we take two points we can calculate the slope

$$S = \frac{y_1 - y_2}{x_1 - x_2} \tag{1}$$

and then the intercept point, with the equation for the line:

$$Y - Y_1 = S(X - X_1) \tag{2}$$

So if a line goes through two points $P = (P_X, P_Y)$ and $Q = (Q_X, Q_Y)$ we can find a new point R.

$$R_Y = (P_Y - Q_Y)/(P_X - Q_X)(R_X - P_X) + P_Y \tag{3}$$

To find R_X I used Maple.
We take the equation for our curve $y^2 = x^3 + ax + b$ and substitute in our equation for R_Y:

$$((P_Y - Q_Y)/(P_X - Q_X)(X - P_X) + P_Y)^2 = X^3 + aX + b \tag{4}$$

Now we solve this equation for X. This gives us the point $-R = (R_X, R_Y)$. We must then negate it to get the final point R.

Adding a point to itself is called "doubling". The formula for the slope has to be changed because we can't divide by zero. We start with the equation of our line and differentiate it to find the slope.

$$D\left\{y^2\right\} = D\left\{x^3 + ax + b\right\} \tag{5}$$

Since y is a function of x we use implicit differentiation to solve this.

$$D\left\{f(x)^2\right\} = 2f(x)D\left\{f(x)\right\} = 2f(x)f'(y) \tag{6}$$

$$D\left\{y^2\right\} = 2yD\left\{y\right\} = 2yy' \tag{7}$$

We find $2yy' = 3x^2 + a$ so therefore $y' = (3x^2 + a)/2y$ which means the slope of our curve when we are doubling is

$$S = (3P_X^2 + a)/2P_Y \tag{8}$$

When we add two points with the same x value, we get the point at infinity, since the slopeof the line between them is zero.

Formulas
These are the formulas that describe the lines and the reflection of points in a finite field. They are the same as those for elliptic curves over the real numbers.

Distinct points: $P + Q = R$ where $P \neq \pm Q$:
$S = (P_Y - Q_Y)/(P_X - Q_X)$
$R_X = S^2 - P_X - Q_X$
$R_Y = S(P_X - R_X) - P_Y$

Doubling $P + P = R$
$S = (3P_X^2 + a)/(2P_Y)$ (where a is the parameter from the elliptic curve)
$R_X = S^2 - 2P_X$
$R_Y = S(P_X - R_X) - P_Y$

To use elliptic curves over fields of characteristic 2 we need slightly different formulas for adding points but it works the same way.

Distinct points: $P + Q = R$ where $P \neq \pm Q$:
$S = (P_Y - Q_Y)/(P_X - Q_X)$
$R_X = S^2 + S + P_X + Q_X + a$
$R_Y = S(P_X + R_X) + R_X + P_Y$

Doubling $P + P = R$
$S = P_X + P_Y/P_X$
$R_X = S^2 + S + a$
$R_Y = P_X^2 + R_X(S + 1)$

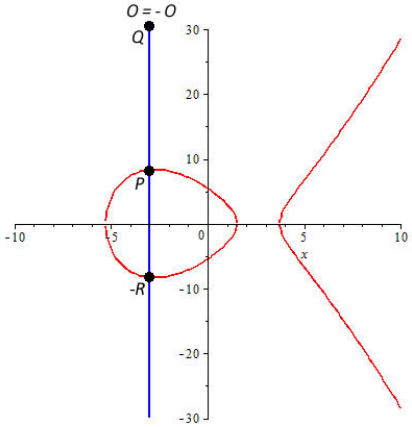
Proving the Properties of Elliptic Curves

Closure

In order for our eliptic curve group to be closed under addition we need to exclude one form of the curve that can occur. If $4a^2 + 27b^2 \neq 0$ the curve will not be closed so we exclude those curves from being used. We formed the equations for the adding of points from all of the possible combinations we could get with the accepted curves. Each of those equations returns a point on the curve or the zero point. Therefore, the set of points is closed under addition.

Additive Identity

We added the zero point to be the additive identity by design. When we add any point and the point at infinity we get back a point that has the same x value but a different y value. When we negate this point it will give us back the original point. Therefore the zero point is our identity.
$P + Q$ gives us $-R$
If $Q = O$ then $P + O$ gives us $-R$
When we negate the point R we get point P. Therefore $P + O = P$
So O is the additive identity for this group



Inverses

Every point (P_X, P_Y) has an inverse $(P_X, -P_Y)$. When we add a point and it's inverse the two points share the same x value but different y values. The slope of the line is zero and we get the point at infinity, which is our additive identity. O is its own inverse.

Addition is Associative
For addition to be associative $(P+Q)+W = P+(Q+W)$.
There are two cases, distinct points and doubling (adding a point to itself). They use the same method but each has it's own set of formulas as seen above. I will only go through the case of distinct points here. These calculations get messy so I used Maple to solve them.

P, Q, W are points on the elliptic curve:
$P = (P_X, P_Y),\ Q = (Q_X, Q_Y),\ W = (W_X, W_Y)$
where we find each y value by plugging x into the curve:

Right hand side: $(P+Q)+W$:
First add P and Q to get point U:
$S1 := (P_Y - Q_Y)/(P_X - Q_X)$
$U_X := S1^2 - P_X - Q_X$
$U_Y := S1 * (P_X - U_X) - P_Y$
Then add U to W to get point V:
$S2 := (U_Y - W_Y)/(U_X - W_X)$
$V_X := S2^2 - U_X - W_X$
$V_Y := S2 * (U_X - V_X) - U_Y$

Left hand side: $P + (Q+W)$
First add Q and W to get point M:
$S3 := (Q_Y - W_Y)/(Q_X - W_X)$
$M_X := S3^2 - Q_X - W_X$
$M_Y := S3 * (Q_X - M_X) - Q_Y$
Then add M to P to get point N:
$S4 := (M_Y - P_Y)/(M_X - P_X)$
$N_X := S4^2 - M_X - P_X$
$N_Y := S4 * (M_X - N_X) - M_Y$

Subtract the x and y values of each side to show they are equal:
$V_X - N_X = 0$ and $V_Y - N_Y = 0$
In Maple these are huge equations but they do cancel each other out in the way I have described. This shows V = N, so addition of points is associative.

Elliptic Curve Public Key Cryptography

Alice and Bob want to send private messages to one another using the elliptic curve group. It requires the same pieces as before but the math is somewhat different. To multiply a point by an integer k, we add the point to itself k times.

These are the pieces needed:
$\mathbb{F}$, a field
C, an elliptic curve
G, generator (a point on the curve)
m, the message (a point on the curve)
a, private key for Alice (an integer)
A = aG, public key for Alice (aG = G + G + G + ...)
b, private key for Bob (an integer)
B = bG, public key for Bob
M = $m + bA$, the encrypted message

Alice and Bob exchange their public keys.

Bob encrypts a message for Alice:

m, message
b, Bob private
A, Alice public
$M = m + bA$

He then sends the encrypted message, M, to Alice.

Alice gets the message M. She then decrypts the message:
a, Alice private
B, Bob public
$m = M - aB = m + baG - abG$

She can then read the unencrypted message m.

Example
Using $\mathbb{F}_8 = \{0, 1, \omega, \omega + 1, \omega^2, \omega^2 + 1, \omega^2 + \omega, \omega^2 + \omega + 1\}$
with the curve $y^2 + xy = x^3 + \omega x^2 + \omega$

Adding Points Table for EC $y^2+wy = x^3 + wx^2 + w$

Points	0 , w^2+w	w^2 , 0	w^2 , w^2	w^2+1 , w+1	w^2+1 , w^2+w	w^2+w+1 , 1	w^2+w+1, w^2+w	O
0 , w^2+w	O	w^2 , w^2	w^2 , 0	w^2+w+1 , w^2+w	w^2+w+1 , 1	w^2+1 , w^2+w	w^2+1 , w+1	0 , w^2+w
w^2 , 0	w^2 , w^2	0 , w^2+w	O	w^2+1 , w^2+w	w^2+w+1 , w^2+w	w^2+1 , w+1	w^2+w+1 , 1	w^2 , 0
w^2 , w^2	w^2 , 0	O	0 , w^2+w	w^2+w+1 , 1	w^2+1 , w+1	w^2+w+1 , w^2+w	w^2+1 , w^2+w	w^2 , w^2
w^2+1 , w+1	w^2+w+1 , w^2+w	w^2+1 , w^2+w	w^2+w+1 , 1	w^2 , w^2	O	0 , w^2+w	w^2 , 0	w^2+1 , w+1
w^2+1 , w^2+w	w^2+w+1 , 1	w^2+w+1 , w^2+w	w^2+1 , w+1	O	w^2 , 0	w^2 , w^2	0 , w^2+w	w^2+1 , w^2+w
w^2+w+1 , 1	w^2+1 , w^2+w	w^2+1 , w+1	w^2+w+1 , w^2+w	0 , w^2+w	w^2 , w^2	w^2 , 0	O	w^2+w+1 , 1
w^2+w+1 , w^2+w	w^2+1 , w+1	w^2+w+1 , 1	w^2+1 , w^2+w	w^2 , 0	0 , w^2+w	O	w^2 , w^2	w^2+w+1 , w^2+w
O	0 , w^2+w	w^2 , 0	w^2 , w^2	w^2+1 , w+1	w^2+1 , w^2+w	w^2+w+1 , 1	w^2+w+1, w^2+w	O

Alice and Bob pick private keys and calculate public keys.
$G = (\omega^2 + 1, \omega + 1)$ Generator point
$a = 5$ Alice
$b = 3$ Bob
$A = aG = G + G + G + G + G = (\omega^2 + \omega + 1, \omega^2 + \omega)$
$B = bG = G + G + G = (\omega^2 + \omega + 1, 1)$

They then exchange the public keys.

Bob encrypts a message:
$m = (\omega^2, \omega^2)$
$b = 3$
$A = (\omega^2 + \omega + 1, \omega^2 + \omega)$
$bA = (\omega^2 + 1, \omega^2 + \omega)$
$M = m + bA = (\omega^2 + 1, \omega^2 + \omega)$

He then sends this to Alice.

Alice decrypts the message:
$a = 5$
$B = (\omega^2 + \omega + 1, 1)$
$aB = (\omega^2 + 1, \omega^2 + \omega)$
$-aB = (\omega^2 + 1, \omega + 1)$
$m = M - aB = m + baG - abG = (\omega^2, \omega^2)$

She can then read the unencrypted message m.

Digital Signatures

A digital signature is a point (r,s) in $\mathbb{F}_n^2$ that is sent along with a message. It proves the message hasn't been altered in transmission, the person who sent it can't deny they sent it and another person can't claim that they were the sender. In the case of cryptocurrencies, the digital signature is the proof of ownership for the coin. The information needed to verify a signature is stored in the transaction logs. When a person wants to claim their coin, they create a signature using their private key, which is then verified by the network before the coin can be spent.

A digital signature generally starts with a complex hash function. It is an algorithm that takes a message of any length and changes it to a message of a specific length while also obscuring the original text. This adds an apparent layer of randomness for additional layer of security. It is a computer science problem, not a math problem so we won't include it here.

These are the main pieces:
$\mathbb{F}$, A finite field
C, An elliptic curve
G, Generator point
n, The order of G (preferably prime)
m, The message, an integer in $\mathbb{F}$
a, The private key, an integer in $\mathbb{F}$
A, The public key (a point)
k, Another integer in $\mathbb{F}$
(r,s), The signature

A practical eliptic curve digital signature would have a hash fuction at the beginning.
$A = aG$
k^{-1} where $k * k^{-1} = 1 \bmod(n)$
$(x_1, y_1) = kG$
$r = x_1 \bmod(n)$
$s = k^{-1}(m + ra) \bmod(n)$
The point (r,s) is our signature. (not a point on the curve)

To verify the signature we need the following pieces:
$w = s^{-1} \bmod(n)$
$u_1 = mw \bmod(n)$
$u_2 = rw \bmod(n)$
$(x_2, y_2) = u_1 G + u_2 A$

If $r = x_2$ then the signature is valid.

Example
$y^2 = x^3 + 5x + 3$ over $\mathbb{F}_{23}$
$G = (1, 3)$
$n = 23$
$m = 5$
$a = 3$
$A = aG = 3(1, 3) = (16, 4)$
$k = 7$
$k^{-1} = 10$

Again we skip the Hash function step.
$(x_1, y_1) = kG = (21, 10)$
$r = x_1 \bmod(23) = 21$
$s = k^{-1}(m + ra) \bmod(23) = 10(5 + 21(3)) \bmod(23) = 13$

This is our signature:
(r,s) = (21,13)

Now we verify the signature.
$w = s^{-1} \bmod(n) = 16$
$u_1 = mw \bmod(n) = 5(16) \bmod(23) = 11$
$u_2 = rw \bmod(n) = 21(16) \bmod(23) = 14$
$(x_2, y_2) = u_1 G + u_2 A = 11(1, 3) + 14(16, 4) = (21, 10)$

In this case $r = 21$ and $x_2 = 21$ so our signature is valid.

Conclusion

From the basic properties of fields to the creation of digital signatures, we have explored eliptic curve cryptography and it's place in cryptocurrencies. The security of these currencies are founded on strong mathematical principles. Knowing the algorithm being used is not enough for an attacker to be able to break the system. They must have the key, which is almost impossible to compute from public information. In a system with a large number of points a person cannot simply reverse engineer the system. The use of group theory gives modern cryptosystems a level of security well beyond anything that could be achieved otherwise. This can then be applied to all sorts of problems in the real world. In cryptocurrencies the ownership of coins is verified in every transaction and they can be safely stored without needing a bank. Elliptic curve cryptography forms the main security that makes that possible.

Sources

Satoshi Nakamoto, *Bitcoin: A Peer-to-Peer Electronic Cash System*, https://bitcoin.org/bitcoin.pdf

Certicom, *ECC Tutorial*, https://www.certicom.com/index.php/ecc-tutorial

Don Johnson, Alfred Menezes, and Scott Vanstone, *The Elliptic Curve Digital Signature Algorithm (ECDSA)*, http://cs.ucsb.edu/ koc/ccs130h/notes/ecdsa-cert.pdf

Crypto-Currency Market Capitalizations, http://coinmarketcap.com/