PUBLIC KEY CRYPTOGRAPHY

ASHWIN RAJAN

1. RSA: A Form of Public Key Cryptography

The main goal of this paper is to talk about the math behind Public Key Cryptography, and what makes it better than symmetrical cryptography.

Definition 1.1. Public key cryptography is an encryption technique that uses a paired public and private key (or asymmetric key) algorithm for secure data communication. [1]

A public key is, as the name suggests, available to the public, but the private key, evidently, is only known to its owner. While the public key and private key must be related mathematically, one cannot efficiently find the private key from a public key (although they can find the public key from the private key). An example of Public Key Cryptography, also known as asymmetrical cryptography, is as follows. There are two people Meriem and Betty. If Meriem wants to send a secret message to Betty, she will encrypt it with the public key, then send the message to Betty. Then, Betty will use her private key to decrypt the message. This method works because if a person, Zachary, intercepts the message along the way, he will have no way to glean the original message, or the private key. One example of public key cryptography is RSA, named after its creators.

Definition 1.2. The RSA assumption is that there is no efficient way to calculate factors of arbitrarily large multiples of primes.

Question 1.3. *So, how does RSA work?*

Answer: Given two selected primes, a computer will calculate $phi(p, q)$, or $(p-1)(q-1)$, and $n = pq$. Using this value, the computer will calculate values e and d. The value e is found by randomly selecting a co-prime of $phi(p, q)$, but less than n. Then, the computer calculates the unique inverse element of e, d, for which $ed \equiv 1 \pmod{phi(p, q)}$. The public key, (e, n), can be shared, but all other values, d, $phi(n)$, p, and q must stay secret. To encrypt a message, all we need is the public key (and the unencrypted message, of course). First, we would translate the letters into numbers, where $A = 65$ and $Z = 90$, with all letters between following consecutively (we will denote this as m). Then, we must find a value o, such that $o \equiv m^e \pmod{n}$. The number o that we find is then translated to a letter (through the same system as above). Whatever letter we arrive at is the letter that is sent to the receiver of this message. We would then continue the process for the rest of the message. To decrypt the message, we need the private key and the encrypted message. First, we translate the letter back to the number, o. To arrive back at the original letter, we must solve the congruence $m \equiv o^d \pmod{n}$. RSA is known for its simplicity, which is why it is so widely used.

Date: December 23, 2018.

[1] `https://www.techopedia.com/definition/9021/public-key-cryptography-pkc`

[2]

Example: Take prime numbers 5 and 7. First, we'll calculate n, which is $pq = 5 * 7 = 35$. Next, we have $phi(p, q)$, which is $(p-1)(q-1) = 4*6 = 24$. Given these values, we can take $e = 19$. To calculate d, we must solve the congruence $19d \equiv 1 \pmod{24}$. Quickly we can solve this by noticing that $19 = -5$, and $(-5)^2 = 25$. As 25 is 1 greater than 24, we have found our inverse to be $-5 = 19$. Thus, we have $(e, n) = (19, 35)$, and $(d, n) = (19, 35)$, [3] our public and private keys, respectively. Now, let's take a message, for example, "Hello World," which translates to 72/69/76/76/79 87/79/82/76/68. Now, we must find an o such that $o \equiv 72^{19} \pmod{35}$. We then get $o \equiv 19467824935503621241553035776019660 8 \pmod{35}$. This is the value sent to the receiver, who then has to decrypt it. Now that we have defined everything, decrypting is relatively simple. All we need now is the private key, $(19, 35)$. All we have to do is solve $m \equiv o^d \pmod{n}$. Once the receiver finds m, they can put it in the form of a letter, and they have the message!

2. Elliptic Curve Cryptography

Definition 2.1. Elliptic Curve Cryptography (ECC) is one of the strongest forms of public key cryptography that depends on the unique properties of elliptic curves.

While completely correct, this definition leads us to another question:

Question 2.2. *What are elliptic curves, and what properties make ECC so strong?*

Answer: Elliptic curves (in Weierstrass normal form) are curves of the form $y^2 = ax^3 + bx + c$, where $4a^3 + 27b^2 \neq 0$ (if it were equal to 0, then the curve would no longer be classified as an elliptic curve; instead it would be a cuspidal curve, or a nodal curve). Elliptic curves also have the property that they are an Abelian Group, which means they satisfy certain requirements. Using this, mathematicians have created Elliptic Curve Cryptgraphy.

Definition 2.3. An Abelian Group is a group with the following properties: a) Closure, b) Associativity, c) Commutativity, d) The existence of an identity element, and e) The existence of an inverse element. [4]

Now, let's go through each of these properties and see how they apply to elliptic curves. (Technically, Elliptic curves are not automatically Abelian Groups; we must first set the parameters for it. Elliptic curves are Abelian groups on rationals, which, as we will see, provides a great benefit to Elliptic Curve Cryptography.) Closure means that if we add (or multiply, but addition is relevant in this case) two elements of the set, we will always arrive at another element in the set. This is true for Elliptic Curves, as, if you add two rational points in an elliptic curve (This is not a normal type of addition, it is changed for the purposes of elliptic curve cryptography. If it helps, you can think of addition as a dot function) you will always get another rational point on the elliptic curve. Associativity has the same definition for groups as it does for arithmetic, so $(p + q) + r = p + (q + r)$. This just means that no matter the order you add points on the elliptic curve, the resultant point will always be the same. Much like associativity, commutativity has the same definition we are used to: $p + q = q + p$. The existence of an identity element is self-explanatory, but in

[2] `http://logos.cs.uic.edu/` (doesn't let me put full url)

[3] You may have noticed that $e = d$ in this case, but that is not necessarily true, as this was merely a coincidence.

[4] `https://crypto.stanford.edu/pbc/notes/group/abelian.html`

elliptic curves, there is no preset identity element. Thus, we have to define one. Here, we will define it to be a point (technically a line, but it acts as one point) defined by $y = \infty$. This designated identity element will now help us define our inverse element. This property means that, given an identity element i, for any point p, there must exist a point q such that $p + q = i$. Replacing i with ∞, we arrive at this property for elliptic curves: $p + q = \infty$.

Now we have the basis for Elliptic Curve Cryptography... except for how to add points.

Definition 2.4. To add two points on an elliptic curve (geometrically), we must first draw a line through both points. The third point it passes through, is then reflected across the x-axis, to get the sum of the two original points.

Well, that definition was kind of confusing, and also didn't give us a clear algebraic method to add two points. Even worse, there are some special cases that were not even mentioned!

First, let's go through the special cases for our definition. Case a) You want to add a point to itself, or multiply a point by two. If we look at

$$\lim_{(x_1,y_1)\to(x_2,y_2)} \frac{(y_2 - y_1)}{(x_2 - x_1)}$$

we find that it is $\frac{df(x)}{dx}$. Thus, if we want to add a point to itself, we can simply take the derivative of the elliptic curve at that point, instead of taking the slope between both points. Case b) The line that goes through both points is of the form $x = a$. Because of this, there is no third point for the line to go through. However, we must remember that there is an identity element we defined previously. This point can be used as a third point if the line is of the form of $x = a$.

Adding two points on an elliptic curve is important, but more important is multiplying a point by an integer. So, building off of the formula for adding two points, we can find a formula for multiply a point by 2, the basis of Elliptic Curve Cryptography. Our formula for the x coordinate of a point 2P (given $P(x, y)$ on curve $y^2 = x^3 + ax + b$) is

$$\frac{x^4 - 2ax^2 - 8bx + a^2}{4(x^3 + ax + b)}. \tag{2.1}$$

Question 2.5. *How does Elliptic Curve Cryptography work?*

Answer: Elliptic Curve Cryptography is, like RSA, a form of public key cryptography. In this case, the private key is a random whole number n, and the public key is a point $Q = nP$. So, if we know the private key, n and P, finding Q is simple, but if we have the public key, P and Q, finding the private key is incredibly difficult (in fact, it requires you to solve the discrete logarithm problem, which for most elliptic curves, is computationally intractable).

3. RSA vs. ECC: which is more effective?

RSA is amazing when you consider it, because of its simplicity and efficiency. However, the bit size for RSA simply cannot match up with the bit size for Elliptic Curve Cryptography. Elliptic Curve Cryptography has extremely low bit sizes for even the largest of primes, while RSA has greater bit sizes. This makes Elliptic Curve Cryptography more efficient, but there is one problem.

Question 3.1. *What is public key cryptography vulnerable to?*

Answer: Cryptography has come such a long way from when we originally started it, and asymmetric (or public key cryptography) methods are far stronger than their symmetric counterparts. However, as we mentioned earlier, if someone can solve the discrete logarithm problem, then everything will come crashing down. Arvind Krishna, director of IBM Research, has made viable claims that he can make a quantum computer in less than 10 years, and modify Shor's algorithm in a way that they could break cryptography forever.

E-mail address: ashwin.rjn793@gmail.com