Elliptic Curves and Cryptography

CHRIS ROHLICEK

May 2, 2018

Introduction

The National Institute of Standards and Technology (NIST) is an agency of the U.S. Department of Commerce whose job today includes the establishment of standards for such practices as the encryption of government information. After Edward Snowden leaked a number of classified documents from the NSA, the means by which that information was supposed to be protected (as dictated by the NIST) came into question. The basis of this data encryption was an algorithm using elliptic curves to generate pseudo-random numbers in such a way that was practically irreversible given the information that was made public. However, it was later discovered that the NSA (who played a large part in the design and standardization of this algorithm) made it possible for them to have access to the information that made this encryption easily reversible. In order to fully understand this story, we will describe the Diffie-Hellman Key Exchange (the encryption algorithm at the root of all the cryptography at play here), the role of elliptic curves in such cryptographic algorithms, and finally the specific mechanism by which the NSA gave themselves the ability to have access to all the information that they helped protect.

Basics of the Diffie-Hellman Key Exchange

One of the most famous advancements in the field of cryptography was made in the 1970s by Whitfield Diffie and Martin Hellman, with the de-

velopment of an algorithm that was the first of its kind to allow for two parties to establish a shared cryptographic key over a public channel of communication. First conceptualized by Ralph Merkle, the Diffie-Hellman algorithm was the first cryptographic tool that did not require two parties to physically meet and establish a shared secret which was necessary for secure communication.

To describe the Diffie-Hellman Key Exchange, we will go through the protocol using elements of a generic group as the vehicle through which the shared secret is established [1]:

Consider two people (called Person A and Person B) who are participating in this exchange. Person A and B both have access to a set of public information, which includes a cyclic group E, that group's order n, and an element $Q \in E$ which generates E. Person A first chooses a random number, e, which they keep secret, but they make public the result they get from multiplying their secret number with the public generator. So now the public information also includes $P = e * Q$. Similarly, Person B comes up with a secret number, r, and releases the public result $r * Q$. Once these preliminary steps are completed, the public pool of information now includes E, n, Q, P, and $r * Q$.

By the commutativity of integers we know the following relation to hold on a subset of the public information:

$$e * (r * Q) = r * (e * Q) = r * P$$

In this relation lies the innovation of the Diffie-Hellman Key Exchange. Because Person A knows their own private number e, as well as the public $r * Q$, they can multiply those together to calculate $r * P$. Similarly for Person B, they can multiply their private r and the public $e * Q = P$ to get $r * P$. Thus $r * P$ is the shared secret between Person A and B.

In this example, $r * P$ is supposed to be a secure secret because we assume that our group E is such that given the information $E, n, Q, P = e * Q, r * Q$, the calculation required to find $r * P$ is intractable. More recently, this

method has been made even more secure by the introduction of elliptic curves as the tool for encryption.

Overview of Elliptic Curves

To understand one of the major tools used in the encryption technology we are discussing, we must first make clear the idea and implications of elliptic curves. We define an elliptic curve over a field F as the set of points satisfying the relation

$$y^2 = x^3 + ax + b$$

As we used in our above illustration of the Diffie-Hellman Key Exchange, we can use the elliptic curve to create a group from which we can take elements to facilitate the establishment of a secret key.

The Group Structure on an Elliptic Curve

Consider the set of all points on an elliptic curve, E. To give this set a group structure we define the operation of addition. The geometric intuition behind addition on an elliptic curve is illustrated below[2]:

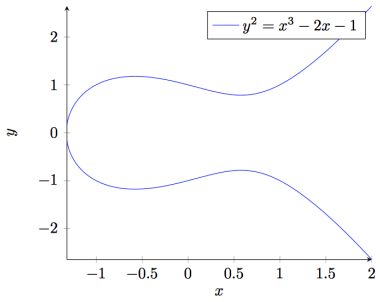

Figure 1: Elliptic Curve over the Real Numbers.

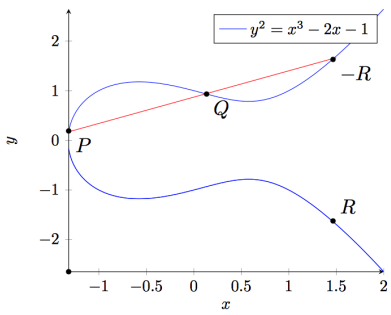

Figure 2: Addition.

Addition

For any two points $P, Q \in E$, we define addition on these points such that $P + Q = -R$. To picture this geometrically, consider the line defined by

going through points P and Q. $-R$ is the third point of contact made with this line and the elliptic curve. To use this point to get our result R, we define R to be the point opposite to $-R$. In other words, for $-R = (x, y)$, $R = (x, -y)$. We know an opposite point will exist for every element of E because an elliptic curve will always be symmetric over the x-axis.

Associativity follows a purely geometric argument through which addition is shown to be associative for any three arbitrary points P, Q, R [3]. Through the addition algorithm described above, one sees that it must hold that $(P + Q) + R = P + (Q + R)$.

Closure

We know that E is closed under addition because for points $P = (x_1, y_1)$ and $Q = (x_2, y_2)$ we can express the coordinates of $R = (x_3, y_3)$ in the following closed form[2]:

$$x_3 = \left(\frac{y_2 - y_1}{x_2 - x_1}\right)^2 - x_2 - x_1$$
$$y_3 = -\left(y_2 + \frac{y_2 - y_1}{x_2 - x_1}(x_3 - x_2)\right)$$

In addition to this, we need to consider the special case in which the points are equal: $P = Q = (x_1, y_1)$. This gives a similar closed form solution for $R = (x_2, y_2)$:

$$x_2 = \frac{3x_1^2 + 1}{2y_1} - 2x_1$$
$$y_2 = -\left(y_1 + \frac{3x_1^2 + a}{2y_1}(x_2 - x_1)\right)$$

Geometrically, when the two points are equal one can imagine finding the third point by looking for the intersection that the tangent line at P has with the curve.

Existence of Inverses and Identity

As we briefly mentioned before, all elements P have an inverse element (equivalent to what we previously referred to as the "opposite") P' such

that $P + P' = 0$. The existence of these inverses follows from the symmetry of the elliptic curve.

The set of elements in E includes all points along the elliptic curve, as well as an extra point $id = (\infty, \infty)$. As you can tell from the notation, this serves as the identity element:

$$\forall P \in E, \;\; P + id = P$$

Point Multiplication

From our definition of addition over our group E, we can extrapolate a notion of scalar multiplication of points. For example, for an integer n, we say that $nP = \underbrace{P + ... + P}_{n terms}$. This is an important feature of our group to consider, because it gives us the ability to form subgroups of E.

Consider the cyclic group generated by a point, P. The cyclic group, (P), is then closed under our operation (addition):

$$nP + mP = \left(\underbrace{P + ... + P}_{n \text{ terms}}\right) + \left(\underbrace{P + ... + P}_{m \text{ terms}}\right) = \left(\underbrace{P + ... + P}_{nm \text{ terms}}\right) = (n + m)P$$

The idea of point multiplication has a very interesting quality in that it is very hard to reverse. As you recall from earlier, addition is defined on our group E in a way that is almost purely geometric, and depends entirely on the shape of a given elliptic curve. For this reason, given points Q and P such that $Q = nP$, there is no simple way to calculate n.

To be more specific about this, consider a subgroup $(P) \subset E$ generated by a point P. Imagine that you were trying to find the order of (P). By Lagrange's theorem we know that the order of (P) must divide the order of E, so if we are given $|E|$ it may seem appealing to fin $|(P)|$ by simply checking all divisors of E until you find the smallest integer n such that $nP = 0$. However, for arbitrarily large E, this can become very difficult. This introduces to us the idea of the discrete logarithm problem.

The Discrete Logarithm Problem

Consider our group E of the points defined by an elliptic curve. For points $P, Q \in E$, the *discrete logaritm* of Q in base P is defined by k such that $Q = kP$. The goal of this problem is to find the smallest such k.

Application to Diffie-Hellman

As we noticed in our summary of the Diffie-Hellman Key Exchange, the security of the algorithm relies on the intractability of solving for either party's secret key given only the public information of a secret key's product with something else. Before we explain the role of elliptic curves in this encryption algorithm, we will once more restate the Diffie-Hellman protocol, but this time in the context of finite cyclic groups (recall that these are the structures of the subgroups on our finite elliptic curves) [4].

- Person A and Person B begin with the shared information G (a finite cyclic group), and g (a generating element).
- Person A picks a number a which becomes their kept secret, and publicly gives the result g^a to Person B.
- Person B picks a number b which becomes their kept secret, and publicly gives the result g^b to Person A.
- With the public information along with their respective private keys, Person A calculates $(g^b)^a$ and Person B calculates $(g^a)^b$. $(g^b)^a = (g^a)^b = g^{ab}$ is then the shared secret which is deemed as such because it is assumed to be intractable to calculate g^{ab} from the public information g, g^a, g^b.

The protocol of this algorithm changes very little when we make elliptic curves the medium for the exchange. Similar to the case above, we begin with a finite cyclic group, which is generated by some point P belonging to the group of points on some elliptic curve over a finite field. With the generator P public, Person A and B then use their secret numbers n_A and n_B respectively to obtain the results $Q_A = n_A * P$ and $Q_B = n_B * P$ which they then publicly send to the other person (this part of the exchange is totally analogous to the non-elliptic curve case, except here we use the

point multiplication defined by our addition operator). Finally, Person A and B use their secret keys to reach the shared secret of

$$n_A * Q_B = n_A n_B * P = n_B n_A * P = n_B * Q_A$$

With a result that looks almost identical to the result that came before the use of elliptic curves, the benefit of this application is not immediately obvious (either result is impractical to reverse, in what way was this an improvement?). Experimentally however, elliptic curve cryptography using 256-bit keys is seen to be equivalent in security to RSA encryption (a method very similar in protocol to Diffie-Hellman but used for some slightly different applications) using 3072-bit keys [5]. The benefit of elliptic curves to this method of encryption is the fact that is presents a vastly more difficult distinct logarithm problem, and is thus vastly more secure.

Application to Random Number Generators

Now that we understand the basics of elliptic curves and their role in cryptography, we will consider a similar application to the technology of pseudo-random number generators. For this technology, the NIST dictates that the following data is public: E and elliptic curve over a finite field (call it $\mathbb{F}_p$), p the order of that field, n the order of the group E, f a cubic polynomial in $\mathbb{F}_p[x]$, and points $P, Q \in E$ [1].

By the nature of points on elliptic curves, we can extract from any non-identity point in E its x-coordinate in the finite field (we will define this extraction with a map ϕ):

$$\phi : (E - \{0\}) \to \mathbb{F}_p$$

Even further, because our field is finite of order p, we know there to exist some injection ψ from $\mathbb{F}_p$ into the integers:

$$\psi : \mathbb{F}_p \to \mathbb{Z}$$

With the composition of these maps, we know there to exist a map by which we can get integers from points on our elliptic curve. This is the main mechanism through which our random numbers are going to be generated.

The method of obtaining random numbers form our set of information (E, p, n, f, P, Q) loosely takes the form of the following algorithm[1]:

- Begin with the integer seed, s, which is kept hidden
- Apply s to the known point $P \in E$, and then map the result to an integer using our composition of functions from earlier: $(\psi \circ \phi)(s * P)$. The use this result to define the integer r.
- Apply r to P and store the result in the integer s': $s' = (\psi \circ \phi)(r * P)$
- Use r and the known point Q to define a separate integer, $t = (\psi \circ \phi)(r * Q)$
- We then convert t to a string of bits and discard the 16 most significant digits to arrive at our random number, b.
- (In the next iteration of this process, s' will be used as the seed).

The Back Door to the Algorithm

As is the case with Diffie-Hellman algorithm, the security of this random number generation comes in the intractable discrete logarithm problem it presents (made especially difficult thanks to the use of elliptic curves). In this case, that problem is solving for a number e such that $P = e * Q$. We know that such an e exists because, since P and Q are both elements of the prime cyclic group E, one must be a multiple of the other.

Ultimately, this random number generating algorithm looses its security once it becomes predictable. To predict the output, one needs to know s', as that is the number that begins the algorithm in the next iteration. One could calculate s' if they knew the product $r * P$, because $s' = (\psi \circ \phi)(r * P)$. As we defined the relation between points P and Q, we know the following equality to hold:

$$r * P = r * (e * Q) = e * (r * Q)$$

From the random number b output by the algorithm, there are then 2^{16} possible t (two possibilities for each of the 16 removed bits from the original t). This is ultimately not a very large number, so it is by no means an

intractable task to check every possibility. For any integer t, there are two possible points on the curve, A and $-A$, for which t represents the $x-$coordinate. Because t is defined by our algorithm as $t = (\psi \circ \phi)(r * Q)$, the proper t is identified when it represents the $x-$coordinate of a point A such that $e * A = r * P$. However, because r is kept unknown, this overarching relation serves as a way to reduce the range of possible values of s', from which the correct one can be found after a relatively small amount of examination of the output of the algorithm.

Conclusion

To understand the cryptography at play in this story, there is only a small set of fundamental ideas with which one must acquaint themselves. We began with the Diffie-Hellman Key Exchange algorithm which was a tremendously innovative addition to the field of cryptography, as it was the first method by which two people could establish a shared secret through only public communication. The reason this was such a groudbreaking algorithm was that the calculations it used to establish this shared secret were practically irreversible. More recently this algorithm was made even more secure by the use of elliptic curves as a basis for the necessary calculations.

With this background made clear we looked at the process by which a good deal of government information is encrypted. However, the NSA, as some of the main architects of this process, gave themselves access to a key piece of information which allowed them to easily undo the encryption algorithm, thus giving themselves access to all of the information which was supposed to be unreachable.

References

1. T.C. Hales. *The NSA Back Door to NIST. Notices of the AMS*. 2014.

2. Jeremy Wohlwend. *Elliptic curve cryptography: pre and post quantum*. 2016.

3. Joseph H. Silverman, John T. Tate. *Rational Points on Elliptic Curves*. Springer-Verlag, New York, 1992.

4. Johannes Buchmann. *Introduction to Cryptography*. Springer-Verlag, New York, 2004.

5. K. Lauter. *The advantages of elliptic curve cryptography for wireless security*. IEEE Wireless Communications, 2004.