To what extent can elliptic curves be used to establish a shared secret over an insecure channel?

Contents

Group Theory

$\mathbb{Z}_p^\times$: The Multiplicative Group Over a Prime

Fermat's Little Theorem suggests the following to be true for any integer a and prime p:

$$a^{p-1} \equiv 1 \pmod{p}$$

Extracting a factor of a, we get

$$a \cdot a^{p-2} \equiv 1 \pmod{p}$$

Thus, under multiplication modulo p, any integer a multiplied by a^{p-2} results in 1. As 1 is the multiplicative identity ($1 \cdot x = x$), a^{p-2} is said to be a's *multiplicative inverse*. Consider the numbers from 1 to $p-1$. Every number has a multiplicative inverse modulo p, and 1 is the identity element as shown above. Further more, multiplication is associative ($a \cdot (b \cdot c) = (a \cdot b) \cdot c$) and each multiplication will always result in a number between 1 to $p-1$ since it is performed modulo p. These properties (existence of an identity element and inverses, associativity and closure of operations) form a group.

We will refer to the group as $\mathbb{Z}_p^\times$. The *order* of a group refers to the number of elements in a group. For $\mathbb{Z}_p^\times$, the order is $p-1$ since the elements are $1, 2, \ldots, p-1$. Using notation, we write $|\mathbb{Z}_p^\times| = p-1$.

The *order* of a specific element x, refers to the smallest integer k such that $x^k = 1$, where 1 is the identity element.[1] For example, the order of 17 in $\mathbb{Z}_{1009}^\times$ is 1008, because 1008 is the smallest integer such that $17^{1008} = 1$, whereas the order 2 in the same group is 504, since $2^{504} = 1$. Therefore, we have $|17| = 1008$ and $|2| = 504$.

The Discrete Log Problem

Under a specific group $\mathbb{Z}_{1009}^\times$, we ask for an integer n for which $17^n = 24$. In this case,

$$17^{456} \equiv 24 \pmod{1009}$$

Therefore $n = 456$ is the solution to this question. More generally, the discrete log problem (DLP) asks for a smallest exponent n for a given group g and its elements a and b such that

$$a^n = b$$

Given this problem, one might take the brute-force or complete search approach, repeatedly performing the group multiplication, calculating a^2, a^3, a^4 and so on to find b. Assuming the exponent n is taken at random, this algorithm would take on average $\frac{1}{2}|a|$ operations. As the order $|a|$ gets big (towards numbers as big as 2^{200}), this approach quickly becomes infeasible.

The assumption that the discrete log cannot be solved trivially in specific groups is the core of cryptographic protocols and algorithms. There are known techniques better than a brute-force search which can solve the discrete log problem either for specific groups or for all groups in general. We shall discuss those methods in a later section, though cryptography is done on well-chosen groups such that even those attacks become infeasible.

[1] On a first glance, the definitions of order for a group and its elements seem to be unrelated. While in fact, the order of an element is also the order of a *subgroup* generated by that element. The meanings of "subgroup" and "generated" are outside the scope of this essay.

Finite Field Cryptography and Attacks

The term *finite field* refers to the fact that the set of numbers from 1 to $p-1$, alongside with zero, form another group under addition. Moreover, multiplication is distributive over addition: $a(b+c) = ab + ac \bmod p$. A field is a set of elements that forms a group under addition and its non-zero elements forms a group under multiplication, where multiplication distributes over addition. The field of integers modulo p is written as $\mathbb{F}_p$.[2]

People may wish to communicate privately over public channels such that the information being transmitted is safe from malicious third-party actors. Encryption protocols using cryptography are in place in internet connections to serve this purpose and defend against attacks. For example, as internet traffic usually goes through a *router*, attackers in public networks such as airport or cafe Wi-Fi can simply pretend to be the router and obtain information if unencrypted.[3]

If the information being communicated is encrypted, then attacks like this would not work since the attacker does not know how to decrypt the information. Common efficient encryption algorithms require the people involved to have a **shared secret**, that is, a password for encrypting and decrypting the messages.[4]

If the only form of communication between two parties is through an **insecure channel**, as in the case of internet connections between someone and their bank, it may be hard to establish such password safely. The Diffie-Hellman Key Exchange is a way to establish a shared secret even if the only channel to communicate in is insecure through the difficulty of the discrete log problem.

Diffie-Hellman Key Exchange

Given a known base x within a group G, one cannot trivially obtain x^{ab} from x^a and x^b if the integers a and b are not known.[5] This is named the Diffie-Hellman problem. If the discrete log problem can be solved trivially, one can simply obtain b from x^b and x, then exponentiate $(x^a)^b = x^{ab}$. As such, the difficulty of the Diffie-Hellman problem in a group is partially related to the difficulty of solving DLP in the same group.

With the Diffie-Hellman problem, Alice can establish a shared secret with Bob by having both generate its own secret exponent - either a or b. Alice can secretly generate a and send Bob g^a, while Bob can secretly generate b and send Alice g^b.

Alice can then compute $\left(g^b\right)^a$ and Bob can compute $(g^a)^b$. As both of these are are equal to g^{ab}, this can be used as the shared secret.

Because only g, g^a, and g^b are sent across the channel, any third party observer will not be able to compute g^{ab} without solving the Diffie-Hellman problem. As we have assumed that the problem is difficult, this is a secure way for Alice to establish a shared secret with Bob over an insecure channel.

---

[2] Note that $|\mathbb{F}_p| = p$ due to the inclusion of zero. We use $\mathbb{F}_p^\times$ to explicitly refer to the multiplicative subgroup where $|\mathbb{F}_p^\times| = p - 1$.

[3] This is called a *man-in-the-middle* attack.

[4] Having a shared secret is called *symmetric key cryptography*. An example of a symmetric key algorithm is the AES

[5] Note that exponentiation here means repeated application of the group operation. In groups where the operation is addition (such as elliptic curves), we will write ax and bx instead.

Index Calculus

Diffie-Hellman Key Exchange on Finite Fields normally uses groups $\mathbb{F}_p$ where $2^{2048} \leq p \leq 2^{8192}$ [1], [2]. The size of the prime ensures that solving DLP is inefficient. Below we will describe Index Calculus, which efficiently solves DLP for smaller finite fields.

It is best to illustrate with an example. We'll reuse the one presented earlier:

$$17^n \equiv 24 \pmod{1009}$$

To find n, we first define a logarithm function L. $L(x)$ is defined such that

$$17^{L(x)} \equiv x \pmod{1009}$$

Note that when we have

$$17^{L(x)+L(y)} \equiv 17^{L(x)} \times 17^{L(y)} \equiv xy \equiv 17^{L(xy)} \pmod{1009}$$

So then

$$17^{L(x)+L(y)-L(xy)} \equiv 1 \pmod{1009}$$

Because the $|17| = 1008$, we have

$$\begin{aligned} L(x) + L(y) - L(xy) &\equiv 0 \pmod{1008} \\ L(x) + L(y) &\equiv L(xy) \pmod{1008} \end{aligned}$$

As such, we have a relation analogous to the laws of logarithm on real numbers. Since every number can be factorized into primes, the idea is the obtain $L(p)$ for small primes p, then figuring out $L(24)$ afterwards. We first try to factorize exponents of the base, 17, looking for ones that can be factorized into relatively small primes:

$$\begin{aligned} 17^{15} &\equiv 2^2 \cdot 5 \cdot 13 \pmod{1009} \\ 17^{16} &\equiv 2^7 \cdot 3 \pmod{1009} \\ 17^{24} &\equiv 2 \cdot 11^2 \pmod{1009} \\ 17^{25} &\equiv 2 \cdot 3 \cdot 13 \pmod{1009} \\ 17^{33} &\equiv 2^2 \cdot 3^2 \cdot 11 \pmod{1009} \\ 17^{36} &\equiv 2^2 \cdot 7^2 \pmod{1009} \end{aligned}$$

Applying L to both sides of the equations, we obtain

$$\begin{aligned} 15 &\equiv 2L(2) + L(5) + L(13) \pmod{1008} \\ 16 &\equiv 2L(7) + L(3) \pmod{1008} \\ 24 &\equiv L(2) + 2L(11) \pmod{1008} \\ 25 &\equiv L(2) + L(3) + L(13) \pmod{1008} \\ 33 &\equiv 2L(2) + 2L(3) + L(11) \pmod{1008} \\ 36 &\equiv 2L(2) + 2L(7) \pmod{1008} \end{aligned}$$

There are six unknowns $L(2), L(3), L(5), L(7), L(11), L(13)$ and six equations, using linear algebra methods, we can arrive at the solution

$$L(2) = 646, L(3) = 534, L(5) = 886,$$
$$L(7) = 380, L(11) = 697, L(13) = 861$$

Next up, the idea is to find $17^x \cdot 24$ and find one that can factorize over primes not greater than 13. Indeed, we have $17^2 \cdot 24 \equiv 2 \cdot 3^2 \cdot 7^2 \pmod{1009}$, so then we have

$$2 + L(24) \equiv L(2) + 2L(2) + 2L(7) \pmod{1008}$$
$$L(24) = 456 = n$$

Therefore, we indeed arrive at the answer $n = 456$. As seen above, this method relies on the property that prime factorizations always exist, which may not apply to ellpitic curves.

General Number Field Sieve[6] is a more sophisticated form of index calculus and in general more efficient than the normal index calculus for large primes [3]. Its time complexity can be given as [4]:

$$\exp\Big((64/9)^{1/3}(\ln p)^{1/3}(\ln \ln p)^{2/3}\Big)$$

where p is the prime that defines the finite field in $\mathbb{F}_p$. Through numerical calculations, we can obtain that in a field where $p = 2^{2048}$ will take 10^{35} times the number of operations as in a field where $p = 3$.

Elliptic Curve Cryptography

Let an elliptic curve be denoted by the equation $y^2 = x^3 + Ax + B$ where A and B are constants. Note that the curve is symmetric about the x-axis, since if (x, y) is a point on the curve, $(x, -y)$ is also on the curve.

Let $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$ be distinct points on the curve, where $x_1 \neq x_2$. We can find a new point on the curve by defining a line that goes across the two points, with slope

$$m = \frac{y_2 - y_1}{x_2 - x_1}$$

And line equation

$$y = m(x - x_1) + y_1$$

We substitute this into the equation of the curve:

$$(m(x - x_1) + y_1)^2 = x^3 + Ax + B$$

Expanding and rearranging gives:

$$x^3 - m^2x^2 + (2m^2x_1 - 2y_1m + A)x + 2y_1mx_1 - m^2x_1^2 - y_1 = 0$$

[6]Name is based on how the factoring step is also done in parallel on a General Number Field. Describing the details of the algorithm requires way more background material than normal index calculus, therefore out of scope of this essay.

With Vieta's formulas, the sum of roots for the cubic is m^2. We already know two roots of this polynomial as P_1 and P_2 are common points on the curve and the line, so we can find the x coordinate of the third point:

$$x_3 = m^2 - x_1 - x_2$$

In the group law, the y coordinate of the resulting point is flipped: (TODO: explain why)

$$y_3 = -(m(x_3 - x_1) + y_1) = m(x_1 - x_3) - y_1$$

Therefore, we have arrived at $P_3 = (x_3, y_3)$, a third point distinct from P_1 and P_2.

If only one point $P_1 = (x_1, y_1)$ is known, we can use implicit differentiation to find the tangent line:

$$y^2 = x^3 + Ax + B$$

$$2y\frac{\mathrm{d}y}{\mathrm{d}x} = 3x^2 + A$$

$$m = \frac{\mathrm{d}y}{\mathrm{d}x} = \frac{3x^2 + A}{2y} = \frac{3x_1^2 + A}{2y_1}$$

With the same line equation $y = m(x - x_1) + y_1$, with the same expanded formula:

$$x^3 - m^2x^2 + \ldots = 0$$

But this time, x_1 is a repeated root, as a tangent line either touches no other points at all (the case when $y = 0$) or touch one other point.

Therefore, we can find the third point with

$$x_3 = m^2 - 2x_1$$

And

$$y_3 = -(m(x_3 - x_1) + y_1) = m(x_1 - x_3) - y_1$$

Therefore, we can begin to define a group law for points on elliptic curves.

Let $C : y^2 = x^3 + Ax + B$ be the elliptic curve with the set of points that satisfy the given equation. We now show that $C \cup \{\infty\}$ forms a group.

Let $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$ be two points that are on the curve. Define $P_3 = P_1 + P_2$ to be as follows:

- If $P_1 = P_2 = (x_1, y_1)$, let

$$P_3 = (m^2 - 2x_1, m(x_1 - x_3) - y_1), \text{where } m = \frac{3x_1^2 + A}{2y_1}$$

- If $x_1 = x_2$ but $y_1 \neq y_2$ (N.B. the only case where this happens is $y_1 = -y_2$): let $P_3 = \infty$.
- Otherwise, let

$$P_3 = (m^2 - x_1 - x_2, m(x_1 - x_3) - y_1), \text{where } m = \frac{y_2 - y_1}{x_2 - x_1}$$

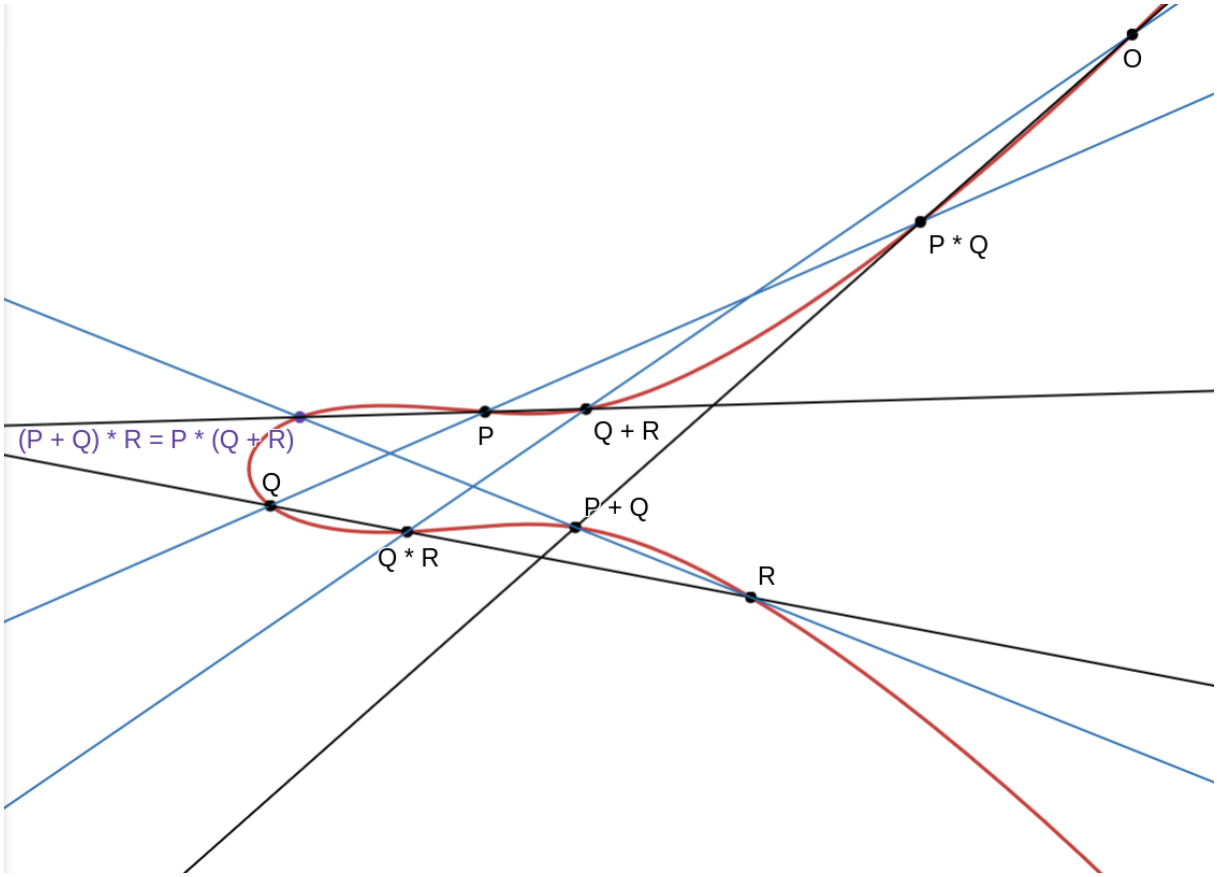
Additionally, define $P_1 + \infty = \infty + P_1 = P_1$, as well as $\infty + \infty = \infty$.

Perhaps the most surprising result of defining this operation is that the operation is associative, that is, $(P_1 + P_2) + P_3 = P_1 + (P_2 + P_3)$ for any three points P_1, P_2, P_3 that belong to the set $C \cup \{\infty\}$. Proving this algebraicly becomes very tedious, but there is a geometric argument using cubics and Bézout's theorem for the case where the points are distinct and none of them have the same x value.

Proof of Associativity

Bézout's theorem states that in general two plane curves given in the equations $a(x, y) = 0$ and $b(x, y) = 0$ with degrees d_a and d_b will have $d_a d_b$ intersections.

Formally, the group of elliptic curves can be defined in projective space where the point at infinity can be treated like any other point. For ease of presentation, the point at infinity will be denote as O.



Note that the intersections between the blue lines and the curve are $(P + Q) * R$, P, Q, R, $Q * R$, $Q + R$, $P * R$, $P + Q$, and O.

The intersections between the black lines and the elliptic curve are $P * (Q + R)$, P, Q, R, $Q * R$, $Q + R$, $P * R$, $P + Q$, and O.

As three lines form a cubic, and the two groups of three lines both intersect the same eight out of the nine points with the elliptic curve, it can be shown that the ninth point is the same for both cubics that intersect with the elliptic curve, thus proving that $(P + Q) * R = P * (Q + R)$.

Group of elliptic curve points

Therefore, $C \cup \{\infty\}$ forms a group since:

1. The operation $+$ is well-defined for any points $P_a + P_b$ where $P_a, P_b \in C \cup \{\infty\}$ as above.
2. ∞ is the identity element, where $P_a + \infty = P_a$ for all $P_a \in C \cup \{\infty\}$.
3. Every element has an inverse: let $P_a = (x, y)$, its inverse is $-P_a = (x, -y)$. We know that $-P_a \in C$ since the curve is given as $y^2 = x^3 + Ax + B$ and swapping y with $-y$ will still hold.
4. The operation is associative.

Since elliptic curve points form a group, cryptographic techniques such as diffie-hellman key exchange which relies on group operations can also be applied to elliptic curves.

Elliptic curve diffie-hellman

One important difference between elliptic curve operations and modular multiplicative group operations is in notation. In elliptic curve, the operation is commonly represented as addition of two points. Therefore $A + B$ is the normal operation on two points A and B while kA is the operation repeated (e.g. $2A = A + A$). In the multiplicative group modulo p, the correspondence goes to AB and A^k. Thus, in previous sections about the multiplicative groups, an operation such as A^k will now be written as kA in the context of elliptic curves.

With that note, diffie-hellman in elliptic curves follows the exact same procedure: two parties agree on a curve group to use, then decide on a base point G. Alice generates a secret integer a and sends Bob aG. Bob generates a secret integer b and sends Alice bG. They can now both calculate abG, which cannot be known by third parties unless they can solve the discrete log problem in elliptic curves.

Finding the Discrete Log with Pollard's ρ algorithm

Pollard's ρ algorithm is a general algorithm for solving the discrete log problem for any abelian group. It is less efficient than the general number field sieve on discrete log in finite fields, taking $O\left(\sqrt{N}\right)$ time on average with N being the order of the group.

We first take an example adapted from page 164 of Silverman and Tate's book: $y^2 = x^3 + 6692x + 9667$, in F_{10037}, with $P = (3354, 7358)$, $Q = (5403, 5437)$. Find k such that $kP = Q$.[7]

Generate 10 random points on the curve based on multiples of P and Q:

$$
\begin{aligned}
M_0 &= 42P + 37Q \\
M_1 &= 21P + 12Q \\
M_2 &= 25P + 20Q \\
M_3 &= 39P + 15Q \\
M_4 &= 23P + 29Q \\
M_5 &= 45P + 25Q \\
M_6 &= 14P + 37Q \\
M_7 &= 30P + 12Q \\
M_8 &= 45P + 49Q \\
M_9 &= 40P + 45Q
\end{aligned}
$$

[7] Originally a Montgomery equation, used substitution to turn it into the short Weierstrass form for consistency.

Then pick, in the same way, a random initial point:

$$A_0 = 15P + 36Q = (7895, 3157)$$

Then, choose an M_i point to add to based on the ones digit of the x coordinate of the point. As A_0 has $x = 7895$, $A_1 = A_0 + M_5 = (7895, 3157) + (5361, 3335) = (6201, 273)$.

Formally, define

$$A_{n+1} = A_n + M_i \text{ where } i \equiv x_n \pmod{10}$$

for $A_n = (x_n, y_n)$. The choice of random M_i points creates a kind of "random walk" of the points in the elliptic curve. As we keep calculating, we get:

$$\begin{aligned} A_0 = (7895, 3157), A_1 &= (6201, 273), ..., \\ A_{95} = (170, 7172), A_{96} &= (7004, 514), ..., \\ A_{100} = (170, 7172), A_{101} &= (7004, 514) \end{aligned}$$

We reach a cycle with $A_{95} = A_{100}$. Since we know the multiples of P and Q for all of the M_i points and thus all A_n points, keeping track of them gives us $A_{95} = 3126P + 2682Q$, we also have $A_{100} = 3298P + 2817Q$. With $3126P + 2682Q = 3298P + 2817Q$, we have:

$$\infty = 172P + 135Q = (172 + 135n)P$$
$$172 + 135n \equiv 0 \pmod{10151} \; n \equiv 1277 \pmod{10151}$$

With verification, we indeed have $1277P = Q$.

Evaluation

Pollard's ρ algorithm works on average with $\sqrt{\frac{\pi}{4}N}$ elliptic curve additions with N being the order for the base point P. [5]

Bibliography

[1] M. Friedl, N. Provos, and W. Simpson, "Diffie-Hellman Group Exchange for the Secure Shell (SSH) Transport Layer Protocol," Mar. 2006, doi: 10.17487/RFC4419.

[2] L. Velvindron and M. D. Baushke, "Increase the Secure Shell Minimum Recommended Diffie-Hellman Modulus Size to 2048 Bits," Dec. 2017. doi: 10.17487/RFC8270.

[3] K. Nguyen, "Index Calculus," *Encyclopedia of Cryptography and Security*. Springer US, Boston, MA, pp. 287–289, 2005. doi: 10.1007/0-387-23483-7_198.

[4] A. K. Lenstra, "L-Notation," *Encyclopedia of Cryptography and Security*. Springer US, Boston, MA, p. 358–359, 2005. doi: 10.1007/0-387-23483-7_237.

[5] D. J. Bernstein, T. Lange, and P. Schwabe, "On the Correct Use of the Negation Map in the Pollard rho Method," in *Public Key Cryptography – PKC 2011*, D. Catalano, N. Fazio, R. Gennaro, and A. Nicolosi, Eds., Berlin, Heidelberg: Springer, 2011, pp. 128–146. doi: 10.1007/978-3-642-19379-8_8.