Diffie-Hellman and Discrete Logarithms

KG

September 26, 2014

Contents

1 Introduction

In this note, we consider the following problem. Alice and Bob wants to establish a shared secret known only by them by communicating via some communications channel. Eve has access to the channel and she may eavesdrop on anything sent over the channel.

Establishing a shared secret is a prerequisite for using the theory of symmetric cryptography to communicate securely over insecure channels. Traditionally, this was done by meeting in person, using couriers or relying on trusted third parties. But

as networks grow and the number of connections increase, the traditional approaches become impractical or introduce unpleasant trust assumptions.

The Diffie-Hellman protocol is a *cryptographic protocol* for establishing a shared secret. Conjecturally, if the underlying mathematical structure is carefully chosen, running the protocol requires Alice and Bob to do relatively little work to establish a shared value, but the eavesdropper Eve will have to do infeasibly much work to deduce the shared value. In other words, the shared value will remain a secret known (fully) only to Alice and Bob.

This note is an introduction to this protocol and the study of its security. Section 2 describes the protocol and its mathematical foundations, namely finite cyclic groups.

As shown in Section 3.1, not every finite cyclic group is suitable for use in the Diffie-Hellman protocol. Section 3 discusses various necessary requirements for a cylic group to be suitable. Two plausible families of cyclic groups based on finite fields and elliptic curves are discussed in Section 4 and 5.

This text is intended for a reader that is familiar with mathematical language, basic algebra (groups, rings, fields and linear algebra) and elementary computer science (algorithms).

This text is sometimes informal, in particular with respect to computational complexity. Every informal claim in this text can be made precise, but the technical details are out of scope for this note.

This text uses colour to indicate who is supposed to know what. When discussing cryptography, red denotes secret information known only by Alice or Bob. Blue denotes information that the eavesdropper will see. Information that is assumed to be known by both Alice and Bob (as well as Eve) is not coloured.

We also colour for theorems about computation, where blue denotes information that an algorithm knows and can use directly, while red denotes information that exists, but has to be computed somehow before it can be used directly. Information that is considered fixed (such as the specific group in use, group order, generator, etc.) is not coloured.

Except for Section 5, we shall write all groups multiplicatively.

2 The Diffie-Hellman Protocol

The *Diffie-Hellman protocol* is a *cryptographic protocol* that allows Alice and Bob to establish a shared value. Alice is the *initiator* in the sense that she sends the first message in the protocol. Bob is the *responder* since he responds to Alice's message.

We first prove that the protocol is *complete*, in the sense that running the protocol without an adversary establishes a shared value. Next, we consider how much work Alice and Bob must do to execute the protocol.

We want to use the protocol to establish a shared *secret*, in the sense that the eavesdropper Eve should not know the shared value. The protocol is based on a finite cyclic group, and the study of its security turns out to involve the study of an interesting computational problem in finite cyclic groups, computing so-called *discrete logarithms*.

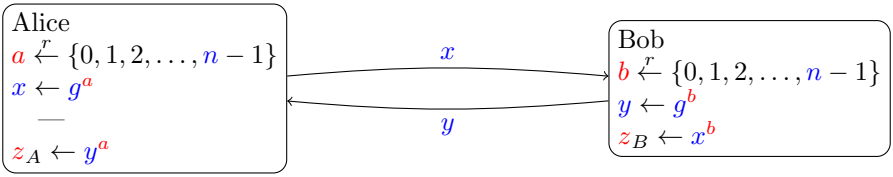


Figure 1: The Diffie-Hellman protocol.

Before we can describe the Diffie-Hellman protocol, we must establish the underlying abstract mathematical structure.

Definition 1. A group G is *cyclic* if there exists an element $g \in G$ such that $G = \{g^a \mid a \in \mathbb{Z}\}$.

Let G be a finite cyclic group of order n with generator g. The Diffie-Hellman protocol works as follows (see also Figure 1).

Both Alice and Bob know the group G, the order of the group n and the generator g.

1. Alice chooses a number a uniformly at random from the set $\{0, 1, 2, \ldots, n-1\}$. She computes $x = g^a$ and sends x to Bob.

2. Bob receives x from Alice. He chooses a number b uniformly at random from the set $\{0, 1, 2, \ldots, n-1\}$, computes $y = g^b$ and $z_B = x^b$, and sends y to Alice.

3. Alice receives y from Bob. Alice computes $z_A = y^a$.

We begin by proving completeness.

Proposition 1. *In the above protocol* $z_A = z_B$.

Proof. We compute that $z_A = y^a = (g^b)^a = (g^a)^b = x^b = z_B$. □

We have shown that the Diffie-Hellman protocol establishes a single shared value, which we shall denote by z.

Next, we must consider how much work Alice and Bob must do to execute the protocol. It is clear that the only non-trivial computations involved are the two *exponentiations* done by each of them. Exponentiation in a group is defined as

$$x^a = \underbrace{x \cdot x \cdots x}_{a \text{ terms}}.$$

From this definition, it is clear that Alice and Bob can do their two exponentiations using less than $2n$ group operations. However, there is a better way to do these computations.

Proposition 2. *For any* $x \in G$ *and non-negative integer* a*, the group element* x^a *can be computed using at most* $2 \log_2 a$ *group operations.*

Proof. We can write

$$a = \sum_{i=0}^{l} a_i 2^i, \qquad a_i \in \{0, 1\}.$$

Since

$$a^{2^{i+1}} = (a^{2^i})^2,$$

we can compute the $l+1$ group elements $x = x^{2^0}, x^{2^1}, x^{2^2}, \ldots, x^{2^l}$ using l group operations.

When these elements have been computed, we can compute the product

$$\prod_{i=0}^{l} (x^{2^i})^{a_i} = x^{\sum_{i=0}^{l} a_i 2^i} = x^a$$

using at most l group operations. Since $l \leq \log_2 a$, the claim follows. □

Exercise 1. The proof of Proposition 2 essentially describes an algorithm for computing x^a. Write out this algorithm carefully and restate Proposition 2 as a statement about the algorithm's time complexity (in terms of group operations, ignoring any other form of computation involved).

Exercise 2. Note that we can define x^a for $a \geq 0$ using the rules that $x^0 = 1$ and

$$x^a = \begin{cases} (x^{a/2})^2 & \text{when } a \text{ is even, and} \\ (x^{(a-1)/2})^2 x & \text{when } a \text{ is odd.} \end{cases}$$

Use this fact to give an alternative proof of Proposition 2 leading to an alternative algorithm.

We have shown that as long as group operations can be computed in reasonable time, Alice and Bob can execute the Diffie-Hellman protocol in reasonable time, even when the group order is very large.

3 Discrete Logarithms

Our goal is for Alice and Bob to establish a shared *secret*. We must consider what the eavesdropper Eve can do to learn the established shared value.

The Diffie-Hellman protocol is supposed to work even when Alice and Bob have had no previous communication. We must therefore assume that Eve knows both the group G, the generator g and the group order n. When Alice and Bob run the protocol, Eve additionally learns x and y. She wants to know z.

Definition 2. The *Diffie-Hellman problem* is to find $z = g^{ab}$ given G, n, g, $x = g^a$ and $y = g^b$, when a and b has been chosen independently and uniformly at random from $\{0, 1, 2, \ldots, n-1\}$.

To use Diffie-Hellman, Alice and Bob need to choose a cyclic group. They want to spend as little effort as possible to establish the shared secret, both with respect to computation and communication. Which means that computing the group operation should be reasonably fast, and group elements should have a reasonably compact representation.

At the same time, Alice and Bob want the established shared value to be secret. It cannot be secret unless solving the Diffie-Hellman problem for the group Alice and Bob use requires more computational effort than Eve can manage.

It is clear that if Eve can find a or b, then she can easily compute $z = y^a = x^b$.

Definition 3. The *discrete logarithm* of x to the base g is the smallest non-negative integer a such that $x = g^a$. We write $\log_g x = a$.

The *discrete logarithm problem* is to find the discrete logarithm of x to the base g, when x has been chosen uniformly at random from the group.

As we see, if Eve can compute discrete logarithms, she can easily compute the shared value established by Alice and Bob. Conversely, it is generally believed (and there is evidence to suggest that this is the case) that if Eve can compute the shared value, she will be able to compute discrete logarithms as well. Under this assumption, the study of the security of the Diffie-Hellman protocol reduces to the study of how easy it is to compute discrete logarithms in the group we want to use.

We first begin with the observation that no choice of generator will make discrete logarithm computations harder.

Exercise 3. Let x be a group element of order m and a, b be integers. Prove that $x^a = x^b$ if and only if $a \equiv b \pmod{m}$.

Proposition 3. *Let g_1 and g_2 be generators, and suppose $\log_{g_1} g_2 = a$. Then $\log_{g_2} g_1 \equiv a^{-1} \pmod{n}$.*

Proof. Suppose $b \equiv a^{-1} \pmod{n}$, that is, $ab = 1 + ln$ for some l. We then compute that

$$g_2^b = (g_1^a)^b = g_1^{ab} = g_1^{1+ln} = g_1.$$

The claim follows by Exercise 3. □

The following exercise will justify that we do not much care about which base we use and often omit it from theorems.

Exercise 4. Show that if you can compute discrete logarithm with some generator g as base, then you can compute discrete logarithms with any generator as base, at roughly twice the cost.

Exercise 5. Any cyclic group G of order n is isomorphic to $\mathbb{Z}_n^+$. Let $\lambda : G \to \mathbb{Z}_n^+$ be a group isomorphism taking a generator g to $1 + \langle n \rangle$. Show that computing discrete logarithms to the base g is essentially the same as computing the group isomorphism λ.

We now begin by establishing a level of effort that will certainly be sufficient to compute discrete logarithms. This will lead to our first requirement for a group to be suitable for Diffie-Hellman.

Proposition 4. *Let G be a cyclic group of order n. The discrete logarithm of a group element $x \in G$ can be computed using less than n group operations.*

Proof. Let g be the generator. Note that if y has discrete logarithm a, then yg has discrete logarithm congruent to $a + 1$ modulo n.

This means that we can compute each element in the sequence $g^0, g^1, g^2, \ldots, g^{n-1}$ using $n - 1$ group operations. Clearly, we can also keep track of the discrete logarithm of each element as we compute it, simply by counting how many group operations we have done.

Since g is a generator, x must be one of the elements in the sequence, and we can recognize it when we reach it. The claim follows. □

The algorithm for compute discrete logarithm implied by the proof of the proposition does more than group operations. But for every algorithm we consider, the group operations will dominate the computational effort required. It therefore makes sense to focus on the number of group operations.

Exercise 6. The proof of Proposition 4 essentially describes an algorithm for computing $\log_g x$. Write out this algorithm carefully and restate Proposition 4 as a statement about the algorithm's time complexity (in terms of group operations, ignoring everything else).

The structure of our study of the discrete logarithm computation is to study various ways of solving or simplifying the computation. Every time we improve our ability to compute discrete logarithms in various groups, we better understand what kind of group Alice and Bob can use for Diffie-Hellman.

Based on Proposition 4, we arrive at the following requirement.

Requirement 1. If n is the group order, n group operations must be an infeasible computation.

3.1 Unsuitable Groups

It is easy to find cyclic groups with large group order. If n is any large number, then $\mathbb{Z}_n^+$ is a cyclic group of order n. A natural generator is $1 + \langle n \rangle$, but the coset of any integer relatively prime to n will do.

Note that this group is written additively. The Diffie-Hellman protocol then looks like:

1. Alice chooses a number a uniformly at random from the set $\{0, 1, 2, \ldots, n - 1\}$. She computes $x = a \cdot (1 + \langle n \rangle) = a + \langle n \rangle$ and sends x to Bob.

2. Bob receives x from Alice. He chooses a number b uniformly at random from the set $\{0, 1, 2, \ldots, n - 1\}$, computes $y = b \cdot (1 + \langle n \rangle)$ and $z = b \cdot x$, and sends y to Alice.

3. Alice receives y from Bob. Alice computes $z = a \cdot y$.

From the description, we see that x is essentially a, and it is immediately obvious that this group is unsuitable for Diffie-Hellman.

Exercise 7. Alice's x is essentially equal to a since we used $1 + \langle n \rangle$ as a generator. Use Proposition 3 to show that any other generator would be equally insecure.

This example shows us that a large group is necessary, but not sufficient for our purposes.

3.2 Pohlig-Hellman I

There are many ways to describe and analyse the first part of the Pohlig-Hellman algorithm for computing discrete logarithms. We shall rely on the algebraic structure of cyclic groups, and begin with the observation that computing a discrete logarithm in G is the same as computing the isomorphism from G to $\mathbb{Z}_n^+$ taking g to $1 + \langle n \rangle$. If G has a suitable group structure, we can use that to simplify the computation of the isomorphism.

Suppose for the remainder of this section that $n = n_1 n_2$ with $\gcd(n_1, n_2) = 1$. Define the two sets

$$H_1 = \{x^{n_2} \mid x \in G\} \text{ and } H_2 = \{x^{n_1} \mid x \in G\}.$$

Exercise 8. Prove that the sets H_1, H_2 are subgroups of G of order n_1, n_2, respectively.

Next, define $\pi_1 : G \to H_1$, $\pi_2 : G \to H_2$ and $\pi : G \to H_1 \times H_2$ by

$$\pi_1(x) = x^{n_2}, \quad \pi_2(x) = x^{n_1} \quad \text{and} \quad \pi(x) = (\pi_1(x), \pi_2(x)).$$

Exercise 9. Prove that the maps π_1, π_2 are well-defined, that they are group homomorphisms and that they are surjective.

Exercise 10. Prove that the map π is a group isomorphism by giving an inverse homomorphism.

It is interesting to compare the above results with the situation for $\mathbb{Z}_n^+$. We begin by stating a version of the Chinese remainder theorem without proof.

Theorem 5. *Let $n = n_1 n_2$ with $\gcd(n_1, n_2) = 1$. Then as rings*

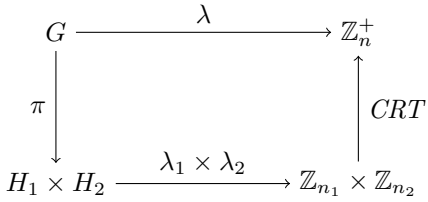
$$\mathbb{Z}_n \simeq \mathbb{Z}_{n_1} \times \mathbb{Z}_{n_2},$$

and the unique ring isomorphism and its inverse are easy to compute.

The map $\mathbb{Z}_n^+ \to \mathbb{Z}_{n_1}^+ \times \mathbb{Z}_{n_2}^+$ corresponding to π above is given by $k + \langle n \rangle \mapsto (kn_2 + \langle n_1 \rangle, kn_1 + \langle n_2 \rangle)$. This map is a group isomorphism, but not a ring isomorphism, unlike the Chinese remainder theorem ring isomorphism which takes $k + \langle n \rangle$ to $(k + \langle n_1 \rangle, k + \langle n_2 \rangle)$.

Let $\lambda : G \to \mathbb{Z}_n^+$, $\lambda_1 : H_1 \to \mathbb{Z}_{n_1}^+$ and $\lambda_2 : H_2 \to \mathbb{Z}_{n_2}^+$ be the group isomorphisms satisfying $\lambda(g) = 1 + \langle n \rangle$, $\lambda_1(\pi_1(g)) = 1 + \langle n_1 \rangle$ and $\lambda_2(\pi_2(g)) = 1 + \langle (n_2 \rangle$. Note that λ_1 and λ_2 are carefully chosen, and correspond to discrete logarithms to the bases $\pi_1(g)$ and $\pi_2(g)$.

Proposition 6. *The diagram*



is commutative. (Note that $\lambda_1 \times \lambda_2$ denotes the group isomorphism taking (x_1, x_2) to $(\lambda_1(x_1), \lambda_2(x_2))$.

Proof. It is sufficient to consider what happens to a group generator.

By definition, we have that $\lambda_1(\pi_1(g)) = 1 + \langle n_1 \rangle$ and $\lambda_2(\pi_2(g)) = 1 + \langle n_2 \rangle$.

Since $\gcd(n_1, n_2) = 1$, the Chinese remainder theorem says that $\mathbb{Z}_{n_1}^+ \times \mathbb{Z}_{n_2}^+$ is isomorphic to $\mathbb{Z}_n^+$. Moreover, the ring isomorphism given by the Chinese remainder theorem takes the ring identity $(1 + \langle n_1 \rangle, 1 + \langle n_2 \rangle)$ to the ring identity $1 + \langle n \rangle$, which concludes the proof. □

Now that we have established the suitable group structure, all we need to do is to ensure that we can use it.

Proposition 7. *Let G be a cyclic group of order $n = n_1 n_2$, $\gcd(n_1, n_2) = 1$. Let H_1 and H_2 be the subgroups of order n_1 and n_2, respectively. The discrete logarithm of any group element $x \in G$ can be computed essentially as fast as we can compute the discrete logarithm of one element in the subgroup H_1 and one element in the subgroup H_2.*

Proof. Consider the diagram from Proposition 6. We can compute the maps π_1 and π_2 quickly by Proposition 2. Computing discrete logarithms in H_1 and H_2 is the same as computing the maps λ_1 and λ_2. Computing the ring isomorphism described by the Chinese remainder theorem can be done quickly.

Computing discrete logarithms in G is equivalent to computing λ. By Proposition 6 and the above argument, λ can be computed such that the only possibly significant cost is computing discrete logarithms in H_1 and H_2. □

Exercise 11. The proof of Proposition 7 essentially describes an algorithm for computing discrete logarithms. Write out this algorithm carefully and restate Proposition 7 as a statement about the algorithm's time complexity (in terms of group operations and computing discrete logarithms in subgroups, ignoring any other form of computation involved).

Theorem 8 (Pohlig-Hellman I)**.** *Let G be a cyclic group of order $n = \prod_{i=1}^{l} \ell_i^{r_i}$, where $\ell_i \neq \ell_j$ when $i \neq j$. The discrete logarithm of a group element $x \in G$ can be computed essentially as fast as we can compute the discrete logarithm of one element in each of the subgroups of order $\ell_i^{r_i}$.*

Proof. We apply Proposition 7 repeatedly and the theorem follows. □

Exercise 12. The proof of Theorem 8 essentially describes an algorithm for computing discrete logarithms. Write out this algorithm carefully and restate Theorem 8 as a statement about the algorithm's time complexity (in terms of group operations and computing discrete logarithms in subgroups, ignoring any other form of computation involved).

You may use the algorithm and statement from Exercise 11 as a subroutine and lemma, respectively.

Based on Proposition 4 and Theorem 8, we arrive at the following requirement.

Requirement 2. If $n = \prod_{i=1}^{l} \ell_i^{r_i}$ is the group order and $\ell_l^{r_l}$ is the largest prime power dividing n, $\ell_l^{r_l}$ group operations must be an infeasible computation.

3.3 Pohlig-Hellman II

In the previous section, we saw how computing discrete logarithms can be reduced to computing discrete logarithms in subgroups whose orders are prime powers. We shall now see how computing discrete logarithms in a group whose order is a prime power can be reduced to computing discrete logarithms in a prime-ordered subgroup.

Suppose for the remainder of this section that the group order n is a prime power ℓ^r for some prime ℓ.

Define the sets

$$H_i = \{x^{\ell^i} \mid x \in G\}, \qquad 0 \leq i \leq r.$$

Note that $H_r = \{1\}$.

Exercise 13. Prove that the sets $H_0, H_1, \ldots, H_r$ are subgroups of G such that $H_0 \supsetneq H_1 \supsetneq \cdots \supsetneq H_r$. Prove also that H_{r-1} is isomorphic to $\mathbb{Z}_\ell^+$.

Now we consider the maps $\pi_i : H_i \to H_{r-1}$ defined by

$$\pi_i(y) = y^{\ell^{r-i-1}}, \qquad 0 \leq i < r-1.$$

Exercise 14. Prove that the π_i maps are surjective group homomorphisms.

Exercise 15. Show that if $y \in H_i$, then $y^{\ell^j} \in H_{i+j}$ for any $j \geq 0$. Show also that $\pi_{i+j}(y^{\ell^j}) = \pi_i(y)$.

Exercise 16. Prove that the kernel of π_i is H_{i+1}.

Proposition 9. *Let g be a generator for G, let $y \in H_i$ and let $a = \log_{\pi_0(g)} \pi_i(y)$. Then*

$$y g^{-\ell^i a} \in H_{i+1}.$$

Proof. Note first that $g^{\ell^i} \in H_i$, and that $\pi_i(y) = \pi_0(g)^a$. Using Exercise 15, we compute

$$\pi_i(y g^{-\ell^i a}) = \pi_i(y)\pi_i(g^{\ell^i})^{-a} = \pi_0(g)^a \pi_0(g)^{-a} = 1.$$

The claim now follows from Exercise 16. □

This suggests a recursive algorithm for computing discrete logarithms.

Theorem 10. *Let G be a cyclic group of order $n = \ell^r$, where ℓ is prime. The discrete logarithm of a group element $x \in G$ can be computed essentially as fast as we can compute r discrete logarithms in the subgroup of order ℓ.*

Proof. We construct a sequence of group elements $y_0, y_1, \ldots, y_{r-1}$ and integers $a_0, a_1, \ldots, a_{r-1}$ as follows. We begin with $y_0 = x$, and compute

$$a_i = \log_{\pi_0(g)} \pi_i(y_{i-1}) \text{ and } y_{i+1} = y_i g^{-\ell^i a_i}, \text{ for } 1 \leq i < r.$$

This requires computing r discrete logarithms in the subgroup H_{r-1}.

By Proposition 9, this sequence is well-defined and $y_r = 1$. Which means that

$$1 = y_{r-1} g^{-\ell^{r-1} a_{r-1}} = \cdots = x g^{-\sum_{i=0}^{r-1} a_i \ell^i}.$$

Note that $0 \leq a_i < \ell$ for $0 \leq i < r$, which means that $0 \leq \sum_{i=0}^{r-1} a_i \ell^i < \ell^r$. By Exercise 3, we get that

$$\log_g x = \sum_{i=0}^{r-1} a_i \ell^i,$$

which concludes the proof. □

Exercise 17. The proof of Theorem 10 essentially describes an algorithm for computing discrete logarithms. Write out this algorithm carefully and restate Theorem 10 as a statement about the algorithm's time complexity (in terms of group operations and computing discrete logarithms in subgroups, ignoring any other form of computation involved).

Based on Proposition 4, Theorem 8 and Theorem 10, we arrive at the following requirement.

Requirement 3. If $n = \prod_{i=1}^{l} \ell_i^{r_i}$ is the group order and ℓ_l is the largest prime dividing n, ℓ_l group operations must be an infeasible computation.

3.4 Shank's Baby-step Giant-step

In this section, we shall improve on Proposition 4 by trading reduced computational effort for increased memory use. Let G be a cyclic group of order n. We must assume that there is some *total order* $\preceq$ on the group elements, such that the following two claims hold when $L \lll n$:

Sorting With effort comparable to computing L group operations, a list of pairs of group elements and integers $(x_1, a_1), (x_2, a_2), \ldots, (x_L, a_L)$ can be rearranged into a new list $(y_1, b_1), (y_2, b_2), \ldots, (y_L, b_L)$ satisfying $y_i \preceq y_j$ when $i \leq j$.

Searching With effort comparable to computing one group operation, we can decide if a group element x is present in a list of pairs of group elements and integers $(y_1, b_1), (y_2, b_2), \ldots, (y_L, b_L)$ satisfying $y_i \preceq y_j$ when $i \leq j$. If the element is present, we also learn what its corresponding number b is.

The algorithm we are interested in begins with the observation that $\log_g x = bL + b'$, where $0 \leq b' < L$.

Theorem 11. *Let G be a cyclic group of order n. For any positive integer $L \lll n$, the discrete logarithm of an element $x \in G$ can be computed using memory to hold L group elements and $L + \lceil n/L \rceil$ group operations.*

Proof. First we construct a list of pairs of group elements and their discrete logarithms

$$(1, 0), (g, 1), (g^2, 2), (g^3, 3), \ldots, (g^{L-1}, L-1).$$

Computing this list requires less than L group operations and memory to hold L group elements. By assumption, we can sort the list into the list

$$(y_1, b_1), (y_2, b_2), (y_3, b_3), \ldots, (y_L, b_L)$$

using effort comparable to what it took to create the list. Note that we can now quickly decide if the discrete logarithm of any group element is less than L, and if so, what its discrete logarithm is.

Recall that we can write $\log_g x = bL + b'$, where $0 \leq b' < L$. Therefore

$$0 \leq \log_g x(G^{-L})^b < L.$$

We can find b and b' by computing successively

$$x, xg^{-L}, x(g^{-L})^2, x(g^{-L})^3, \ldots$$

and for each element check if it is in our list. Note that computing the next element requires one group operation, while the check requires comparable effort.

We know that we will find an element in our list before computing $\lceil n/L \rceil$ elements. It follows that we will find the discrete logarithm of x with at most $L + \lceil n/L \rceil$ group operations. □

Exercise 18. The proof of Theorem 11 essentially describes an algorithm for computing discrete logarithms. Write out this algorithm carefully and restate Theorem 11 as a statement about the algorithm's time complexity (in terms of group operations, ignoring any other form of computation involved).

Exercise 19. What value of L minimizes computational effort?

Exercise 20. Decide the optimal value of L if we need to compute the discrete logarithm of k group elements $x_1, x_2, \ldots, x_k$.

Exercise 21. Suppose we know that $\log_g x < k$. Show that for any $L > 0$ we can compute this discrete logarithm using at most $L + \lceil k/L \rceil$ group operations.

Exercise 22. Suppose we know that $k_1 \leq \log_g x < k_2$. Show that for any $L > 0$ we can compute this discrete logarithm using at most $L + \lceil (k_2 - k_1)/L \rceil$ group operations.

Based on Theorems 8, 10 and 11, we arrive at the following requirement.

Requirement 4. If n is the group order and ℓ is the largest prime dividing n, then $L + \lceil \ell/L \rceil$ group operations using memory for L group elements must be an infeasible computation.

3.5 Pollard's ρ

In this section, we shall consider a group G with a prime n number of elements. In the previous section, we saw that we could trivially recover the discrete logarithm of x given an equation of the form $x(g^{-L})^b = g^{b'}$. Pollard's ρ method will try to generate an equation of the form

$$g^a x^b = g^{a'} x^{b'}, \tag{1}$$

with $b \not\equiv b' \pmod{n}$, after which it is easy to see that $\log_g x \equiv (a - a')(b' - b)^{-1} \pmod{n}$.

Pollard's ρ method relies on connecting two separate ideas with a conjecture to arrive at an algorithm for finding a relation like (1) above. The first idea is that in sequences of randomly chosen elements we will see repetitions fairly soon. The second idea is that in certain sequences we can quickly find cycles. The conjecture is that, with respect to repetitions, certain non-random sequences "look random". It is the repetitions in this sequence that will give us the required relation.

We begin with the first idea and a sequence of elements chosen at random. What is the probability that there are no repetitions within the first L elements of the sequence?

Proposition 12. *Suppose we have a sequence of elements $s_1, s_2, s_3, \ldots$, the elements chosen independently and uniformly at random from a set S with n elements. Let E be the event that the L first elements are all distinct. Then*

$$1 - \frac{L(L-1)}{2n} \leq \Pr[E] \leq \exp\left(-\frac{L(L-1)}{2n}\right).$$

Proof. If $L \geq n$, then $\Pr[E] = 0$ and the claim holds. So we may assume $L < n$.

Choosing elements one after another, we see that we have n choices for the first element, $n-1$ choices for the second element, $n-2$ choices for the third element, and so forth. By independence and uniformity, we get that

$$\Pr[E] = (1 - 1/n)(1 - 2/n) \cdots (1 - (L-1)/n).$$

Note that $1 - \epsilon \leq \exp(-\epsilon)$ for any ϵ, so

$$\Pr[E] \leq e^{-1/n} e^{-2/n} \cdots e^{-(L-1)/n} = \exp\left(-\sum_{i=1}^{L-1} i/n\right) = \exp\left(-\frac{L(L-1)}{2n}\right).$$

For the lower bound, we consider the complementary event. Let F_i be the event that the ith chosen element coincides with at least one of the previous elements. Then $\Pr[F_i] \leq (i-1)/n$, and we get that

$$\begin{aligned} 1 - \Pr[\neg E] &= 1 - \Pr[F_1 \vee F_2 \vee \cdots \vee F_L] \\ &\geq 1 - \sum_{i=1}^{L} \Pr[F_i] \geq 1 - \sum_{i=1}^{L} \frac{i-1}{n} \geq 1 - \frac{L(L-1)}{2n}, \end{aligned}$$

which concludes the proof. □

Note that the sequence we shall consider later is far from random, and this proposition and its proof do not apply. But many non-random sequences are still "random-looking" with respect to repetitions in the sequence, which means that the bounds in the proposition apply in practice. This will be sufficient for our purposes.

Next, we consider a special kind of infinite sequences generated by a starting point $s_1 \in S$, a function $f : S \to S$ and the rule $s_{i+1} = f(s_i)$. The sequence eventually becomes cyclic (for some integers i, j, k, $s_{j+k} = s_j$ when $j \geq i$) when S is finite.

Proposition 13. *Let $s_1, s_2, \ldots$ be the sequence defined by s_1 and the rule $s_{i+1} = f(s_i)$. Suppose k is the smallest integer such that $s_k = s_{k'}$ for some $k' < k$. Then distinct indexes i, j can be found such that $s_i = s_j$ using at most $3k$ evaluations of f.*

Proof. We consider the sequence $t_1, t_2, \ldots$ given by $t_j = s_{2j}$. It is clear that for some i, $t_i = s_i$, and that this i is at most k.

We can compute successively the pairs $(s_1, t_1), (s_2, t_2), \ldots$ using the rule

$$(s_{i+1}, t_{i+1}) = (f(s_i), f(f(t_i))).$$

We will notice when $s_i = t_i$, at which point we have found $s_i = s_{2i}$. Computing each new pair requires evaluating f three times. The claim follows. □

Exercise 23. The proof of Proposition 13 essentially describes an algorithm for finding two integers. Write out this algorithm carefully and restate Proposition 13 as a statement about the algorithm's time complexity (in terms of evaluations of the function f).

Now we are ready to construct a sequence that should allow us to find an equation like (1) and thereby compute discrete logarithms.

Our set will be the group G, of course. Suppose $\{S_1, S_2, S_3\}$ is a partition of G – S_1, S_2 and S_3 are pairwise disjoint sets whose union is G – where the three subsets have approximately the same cardinality. Suppose also that it is easy to check wich subset an element is in.

Now we construct a sequence $y_1, y_2, \ldots$ based on S_1, S_2, S_3, a generator g and a group element x. We let $y_1 = x$ and

$$y_{i+1} = \begin{cases} y_i g & y_i \in S_1, \\ y_i^2 & y_i \in S_2, \text{ or} \\ yx & y_i \in S_3. \end{cases} \tag{2}$$

Based on the sequence $y_1, y_2, \ldots$, we can define a sequence of integer pairs $(a_1, b_1), (a_2, b_2)$, starting with $(a_1, b_1) = (1, 0)$ and using the rule

$$(a_{i+1}, b_{i+1}) = \begin{cases} (a_i + 1 \bmod n, b_i) & y_i \in S_1, \\ (2a_i \bmod n, 2b_i \bmod n) & y_i \in S_2, \text{ or} \\ (a_i, b_i + 1 \bmod n) & y_i \in S_3. \end{cases} \tag{3}$$

Exercise 24. Show that the above two sequences defined by (2) and (3) satisfy $y_i = g^{a_i} x^{b_i}$.

Exercise 25. Let $y_1, y_2, \ldots$ and $(a_1, b_1), (a_2, b_2), \ldots$ be the above sequences defined by (2) and (3) and suppose k is the smallest integer such that $y_k = y_{k'}$ for some $k' < k$. Prove that distinct indexes i, j along with corresponding pairs (a_i, b_i) and (a_j, b_j) can be found such that $y_i = y_j$ using at most $3k$ group operations and $6k$ additions and multiplications modulo n.

Let E be the event that the L first elements of $y_1, y_2, \ldots$ are all distinct, and let E' be the event that the L first pairs of $(a_1, b_1), (a_2, b_2), \ldots$ are all distinct. Then we can define the two functions

$$\theta(L, n) = \Pr[E] \qquad \text{and} \qquad \gamma(L, n) = 1 - \Pr[E'].$$

We can now prove the following result.

Theorem 14. *Let G be a cyclic group of order n. The discrete logarithm of an element $x \in G$ can be computed using $3L$ group operations and $6L + 3$ arithmetic operations modulo n, except with probability at most $\theta(L, n) + \gamma(L, n)$.*

Proof. Some element will appear twice among the $2L$ elements of the sequence described by (2) except with with probability $\theta(L, n)$.

The corresponding pairs (a_i, b_i) and (a_j, b_j) will be distinct except with probability $\gamma(L, n)$.

This repetition $y_i = y_j$ gives us an equation of the form (1), namely

$$g^{a_i} x^{b_i} = g^{a_j} x^{b_j}.$$

When $b_i \neq b_j$, we can compute the discrete logarithm of x, since n is prime.

By Exercise 25 we can find the indexes of the repetition, while at the same time keeping track of the sequence described by (3), using at most $3L$ group operations and $6L$ arithmetic operations modulo n. The claim follows. □

Exercise 26. The proof of Exercise 25 essentially describes an algorithm for finding six integers. Write out this algorithm carefully and restate the claim in Exercise 25 as a statement about the algorithm's time complexity (in terms of group operations and arithmetic operations modulo n).

When we choose a reasonable partition $\{S_1, S_2, S_3\}$, it seems plausible that something similar to the claims of Proposition 12 should hold for the two sequences, and in practice, this seems to be true. We phrase this in terms of a conjecture. Note that in this conjecture the probability is taken over the choice of the element x.

Informal conjecture 15. *For reasonable partitions $\{S_1, S_2, S_3\}$, the function $\theta(L, n)$ is roughly similar to*

$$\exp\left(-\frac{L(L-1)}{2n}\right)$$

and $\gamma(L, n)$ *is roughly similar to*

$$1 - \frac{L(L-1)}{2n^2}.$$

Based on Conjecture 15 and Theorems 8, 10 and 14, we arrive at the following requirement.

Requirement 5. If n is the group order and ℓ is the largest prime dividing n, then $\sqrt{\ell}$ group operations must be an infeasible computation.

4 Finite Fields

The non-zero elements of a finite field with q form an abelian group under multiplication, and we denote this group by $\mathbb{F}_q^*$. The question we shall investigate is if this group is suitable for use in Diffie-Hellman.

We begin by showing that this group is cyclic.

Proposition 16. *The group* $\mathbb{F}_q^*$ *is cyclic.*

Proof. Let n be the maximal order of any element in $\mathbb{F}_q^*$. We know that the order of any element in a group divides the group order, so specifically $n \leq q - 1$

For any $x \in \mathbb{F}_q^*$ it is easy to see that $x^n = 1$ or $x^n - 1 = 0$.

Now consider the polynomial $X^n - 1$. We know that a polynomial of degree $d > 0$ over any field has at most d zeros. As we have just seen, all of the $q - 1$ elements of $\mathbb{F}_q^*$ are zeros of this polynomial, so $n \geq q - 1$. We can conclude that $n = q - 1$.

Since we have an element of order $q - 1$, which is also the group order, the group is cyclic. □

Requirement 5 says that the order of any group we use in Diffie-Hellman should be divisible by a large prime. In other words, we need a large prime power q such that $q - 1$ is divisible by a large prime.

It turns out (for reasons that we shall not investigate) that computing discrete logarithms in extension fields is easier than in prime fields of comparable size. Therefore we restrict our study to prime fields.

The prime number theorem says that prime numbers are fairly common, so one strategy we could try was to find one large prime p such that $2p + 1$ is prime too. Such a prime is called a *Sophie-Germain* prime, and in practice they seem to be so common that if we just choose numbers at random, we will run into a Sophie-Germain prime within reasonable time. But the question is how we will know when we have run into a Sophie-Germain prime.

4.1 Group Operation

Let p be a prime. Mathematically, the finite field $\mathbb{F}_p$ consists of a set of p elements, two of which are distinguished, along with two binary operations $+$ and $\cdot$.

The field is isomorphic to the factor ring $\mathbb{Z}/\langle p \rangle$, which is often a convenient to compute in. To add, subtract or multiply, we simply add, subtract or multiply representatives of the cosets to get new representatives. To divide by $\xi + \langle p \rangle$, we first find an inverse ζ of ξ modulo p, then multiply by $\zeta + \langle p \rangle$.

This is unproblematic mathematically, but computationally it is awkward because the size of the representatives tends to grow very quickly, making arithmetic very slow. However, we know that ξ and ζ represent the same coset if and only if they are congruent modulo p. After the first arithmetic operation, we replace the representative by its remainder when divided by p.

What happens is that we represent the field elements using the integers $\{0, 1, \ldots, p-1\}$ and do arithmetic as integer arithmetic followed by taking the remainder after division by p.

It follows that a group operation in costs two arithmetic operations, while finding an inverse in the group costs three arithmetic operations.

4.2 Primality Testing

Throughout this section, we shall take n to be a large odd integer.

We want to be able to distinguish prime integers from composite integers. The following proposition is obvious, since a composite integer must have a proper divisor smaller than the square root of the integer and we can in principle check every possible divisor.

Proposition 17. *We can decide if a number n is prime using at most $\sqrt{n}$ integer divisions.*

Unfortunately, the algorithm implied by this proposition is useless for large integers, and therefore for our purposes.

Having divisors is one difference in behaviour between composite and prime integers. Another potential difference is given by Fermat's little theorem.

Theorem 18. *If n is prime, then for any integer a not divisible by n,*

$$a^{n-1} \equiv 1 \pmod{n}. \tag{4}$$

It is easy to find composite numbers for which this theorem does not hold, and this suggest a rather simple algorithm that may prove that a number is not prime. Choose a random integer a between 1 and n and compute $a^{n-1} \bmod n$. If the result is not 1, we know that n cannot be prime, and we say that a is a *witness* to this fact.

What if the result is 1? We do not then know that n is prime, since the relation (4) holds for many composite numbers n and choices of a. But as we shall see, we may have some evidence that n is prime.

Exercise 27. Let $G_n \subseteq \mathbb{Z}_n^*$ be the set

$$G_n = \{x \in \mathbb{Z}_n^* \mid x^{n-1} = 1\}. \tag{5}$$

Show that G_n is a subgroup of $\mathbb{Z}_n^*$, and that if n is prime then $G_n = \mathbb{Z}_n^*$.

Exercise 28. Show that there is an algorithm that can decide if an element $x \in \mathbb{Z}_n^*$ is in G_n or not using at most one exponentiation.

Proposition 19. *Let n be an integer and let G be a proper subgroup of $\mathbb{Z}_n^*$. The probability that k elements chosen independently and uniformly at random from $\mathbb{Z}_n^*$ all are in G is at most 2^{-k}.*

Proof. The order of G divides the order of $\mathbb{Z}_n^*$, so $|G|/|\mathbb{Z}_n^*| \leq 1/2$. This means that the probability that any one element is in G is at most $1/2$. Independence then says that the probability that all of them are in G is at most $1/2^k$. □

Exercise 29. Exercise 28 and the proof of Proposition 19 can be combined into an algorithm that decides if the subgroup G_n is a proper subgroup of $\mathbb{Z}_n^*$. Write out this algorithm carefully and formulate and prove a statement about the algorithm's time complexity (in terms of exponentiations) and success probability (the probability that the algorithm decides correctly).

With Exercise 29 we have an algorithm for deciding if G_n equals $\mathbb{Z}_n^*$ with a very small error probability. By Exercise 27 we know that if the two groups are distinct, then n is composite. But what if the two groups are the same? Is n prime?

Definition 4. A *Carmichael number* is a composite integer n such that $G_n = \mathbb{Z}_n^*$.

Unfortunately, Carmichael numbers not only exist, but there are many of them. Which means that the algorithm from Exercise 29 does not distinguish prime numbers from composite numbers, but rather prime and Carmichael numbers from other numbers. This is insufficient for cryptographic purposes.

We can consider the Jacobi symbol $\left(\frac{\cdot}{n}\right)$ to be a map from $\mathbb{Z}_n^*$ to its subgroup $\{\pm 1\}$. Note that when n is prime, the Jacobi symbol conincides with the Legendre symbol, and we have the following theorem.

Theorem 20. *Let n be an odd prime and let $a \in \mathbb{Z}_n^*$. Then*

$$\left(\frac{a}{n}\right) = a^{\frac{n-1}{2}}. \tag{6}$$

Proof. Recall that the Legendre symbol of a group element is 1 if and only if that element is a square in the group.

When n is prime, Proposition 16 says that $\mathbb{Z}_n^*$ is cyclic of order $n-1$. Let g be any generator. When the group order $n-1$ is even, the squares are the elements of the form g^{2i} for integers i, while the non-squares are of the form g^{2i+1}. We see that

$$(g^{2i})^{\frac{n-1}{2}} = (g^{n-1})^i = 1 \qquad \text{and} \qquad (g^{2i+1})^{\frac{n-1}{2}} = (g^{n-1})^i g^{\frac{n-1}{2}} = -1,$$

which proves the theorem. □

Exercise 30. Let $H_n \subseteq \mathbb{Z}_n^*$ be the set

$$H_n = \left\{ x \in \mathbb{Z}_n^* \mid x^{\frac{n-1}{2}} = \left(\frac{x}{n}\right) \right\}$$

Show that H_n is a subgroup of $\mathbb{Z}_n^*$, and that if n is prime then $H_n = \mathbb{Z}_n^*$.

Exercise 31. Exercise 30 and the proof of Proposition 19 can be combined into an algorithm that decides if the subgroup H_n is a proper subgroup of $\mathbb{Z}_n^*$. Write out this algorithm carefully and formulate and prove a statement about the algorithm's time complexity (in terms of exponentiations) and success probability. Note that the Jacobi symbol can be computed with significantly less effort than an exponentiation in the group.

It follows that if we can prove that H_n equals $\mathbb{Z}_n^*$ if and only if n is prime, then we have an algorithm for deciding if a number is prime or not.

Theorem 21. *Let n be an odd composite number. Then there exists $a \in \mathbb{Z}_n^*$ such that* (6) *does not hold.*

Proof. We consider two cases, whether or not n is square-free.

First, suppose there is a prime p such that p^2 divides n. Let $n_2 = n/p$, and let a be the integer $1 + n_2$. The Jacobi symbol is

$$\left(\frac{a}{n}\right) = \left(\frac{a}{p}\right)\left(\frac{a}{n_2}\right) = \left(\frac{1}{p}\right)\left(\frac{1}{n_2}\right) = 1.$$

We shall show that a has order p modulo n. Since p does not divide $n-1$, it will follow that $a^{(n-1)/2}$ is not congruent to 1 modulo n, and therefore that $a+\langle n\rangle$ does not satisfy (6).

From the binomial theorem, it follows that

$$a^p \equiv (1+n_2)^p \equiv \sum_{i=0}^{p} \binom{p}{i} n_2^i \pmod{n}.$$

It is obvious that n divides every other term of the sum except the first term (which is 1) and possibly the second term. But since p divides $\binom{p}{1}$, n divides the second term too, and a^p has order p modulo n.

Next, suppose that no square of any prime divides n. Let p be any prime dividing n and let $n_2 = n/p$. Let also b be any integer that is not a square modulo n with $\gcd(b, n) = 1$. By the Chinese remainder theorem, we can find an integer a satisfying

$$\begin{aligned} a &\equiv 1 \pmod{n_2}, \\ a &\equiv b \pmod{p}. \end{aligned}$$

Note that the second equation means that

$$a^{(n-1)/2} \equiv b^{n-1)/2} \equiv 1 \pmod{n_2},$$

which means that $a^{(n-1)/2} \not\equiv -1 \pmod{n}$. Finally, we compute the Jacobi symbol of a as

$$\left(\frac{a}{n}\right) = \left(\frac{a}{p}\right)\left(\frac{b}{n_2}\right) = (-1)\cdot 1 = -1.$$

It follows that (6) does not hold for $a + \langle n\rangle$. □

It follows that the algorithm from Exercise 31 can efficiently decide if some integer is prime or not.

4.3 Index Calculus

All of the algorithms from Section 3 will work for $\mathbb{F}_p^*$. But it turns out that we can do very much better. We shall develop the ideas of index calculus in a general setting, and then show how the properties of prime fields allow us to apply the ideas to get a more efficient algorithm for computing discrete logarithms.

We begin with an observation about abstract cyclic groups, which is an extension of (1). Let G be a cyclic group of order n. Let g be some generator and let x be a group element. Suppose we have ν pairs of integers $(r_1, t_1), (r_2, t_2), \ldots, (r_\nu, t_\nu)$ and integers $\alpha_1, \alpha_2, \ldots, \alpha_\nu$ such that

$$\prod_{i=1}^{\nu} \left(x^{t_i} g^{r_i}\right)^{\alpha_i} = 1. \tag{7}$$

This will give us the equation

$$x^{\sum_i \alpha_i t_i} g^{\sum_i \alpha_i r_i} = 1,$$

which is of the same form as (1). As long as $\sum_i \alpha_i t_i$ is invertible modulo n, we can recover $\log_g x$ from the equation using $2\nu+2$ arithmetic operations ($\nu+1$ multiplications, ν additions and one inversion).

We first consider the case when the group order n is prime.

Exercise 32. Suppose n is prime. Let

$$y_i = x^{t_i} g^{r_i},\ i = 1, 2, \ldots, \nu.$$

Suppose further that the coefficients $r_1, r_2, \ldots, r_\nu$, $t_1, t_2, \ldots, t_\nu$ have been chosen uniformly at random, but the coefficients $\alpha_1, \alpha_2, \ldots, \alpha_\nu$ only depend on the group elements $y_1, y_2, \ldots, y_\nu$. Show that $\sum_i \alpha_i t_i$ is divisible by n with probability $1/n$.

Proposition 22. *Suppose n is prime and that $\bar{\ell}_1, \bar{\ell}_2, \ldots, \bar{\ell}_l$ are elements of G. Given $l+1$ distinct, non-trivial relations of the form*

$$x^{t_i} g^{r_i} = \prod_{j=1}^{l} \bar{\ell}_j^{s_{ij}}, \qquad i = 1, 2, \ldots, l+1, \tag{8}$$

we can compute $\alpha_1, \alpha_2, \ldots, \alpha_{l+1}$ satisfying (7) using at most $(l+1)^3$ arithmetic operations.

Proof. Let S be the $(l+1) \times l$ matrix (s_{ij}), where each of the relations defines one row. If we consider S as a matrix over $\mathbb{F}_n$, it has rank at most l, so there exists a vector $\vec{\alpha}$ such that $\vec{\alpha} S = \vec{0}$. Given such a vector, we get

$$\prod_{i=1}^{l+1} (x^{t_i} g^{r_i})^{\alpha_i} = \prod_{i=1}^{l+1} \left(\prod_{j=1}^{l} \bar{\ell}_j^{s_{ij}} \right)^{\alpha_i} = \prod_{j=1}^{l} \bar{\ell}_j^{\sum_{i=1}^{l+1} \alpha_i s_{ij}} = 1.$$

Gaussian elimination will find a vector in the kernel of S using at most $(l+1)^3$ arithmetic operations. □

Proposition 23. *Suppose n is prime and that $\bar{\ell}_1, \bar{\ell}_2, \ldots, \bar{\ell}_l$ are elements of G. Suppose further that for a group element y chosen uniformly at random from G we can find a relation of the form*

$$y = \prod_{j=1}^{l} \bar{\ell}_j^{s_j} \tag{9}$$

with probability σ, using at most τ arithmetic operations.

Then, except with probability $1/n$, we can compute $\log_g x$ using an expected

$$\sigma^{-1}(l+1)(\tau + 2\chi) + (l+1)^3 + 2l + 3$$

arithmetic operations, where χ is the number of arithmetic operations required for an exponentiation in G.

Proof. We shall choose random group elements and for each element use at most τ arithmetic operations to try to find a relation of the form (9). When we have found $l+1$ relations, we shall stop.

We choose random elements of the form $x^t g^r$, ensuring that the coefficients s_j will be independent of the exact random numbers t, r chosen, satisfying the requirements of Exercise 32.

Now we can apply Proposition 22 to compute a relation among the random group elements, which by Exercise 32 will allow us to compute the discrete logarithm except with probability $1/n$.

We expect to find $l+1$ relations after choosing $\sigma^{-1}(l+1)$ random group elements.

Sampling a random element costs 2χ arithmetic operations, and trying to find relations costs τ arithmetic operations per group element tested.

The linear algebra will require $(l+1)^3$ arithmetic operations, and computing $\log_g x$ will cost at most $2l+3$ arithmetic operations. The claim follows. □

If n is a prime power q^k then Proposition 22 does not apply. The following exercise suggests how this can be remedied.

Exercise 33. Suppose n is a prime power q^k. Consider the matrix S derived from the relations in the proof of Proposition 22. Suppose that S has rank l modulo q. Show how we can compute $\alpha_1, \alpha_2, \ldots, \alpha_{l+1}$ satisfying (7) except with probability $1/q$ using at most $(l+1)^3$ arithmetic operations.

If q is small, the algorithms from Section 3 suffice to compute the discrete logarithm quickly. Suppose therefore that q is not small. If the elements $\bar{\ell}_1, \bar{\ell}_2, \ldots, \bar{\ell}_l$, the set $\bar{P}$ and the method for generating relations have been chosen such that the relations generated tend not to be systematically linearly dependent, we can reasonably expect that the matrix S has maximal rank.

We therefore know how to deal with groups of prime power order. Finally, we consider the case of a general group order.

Proposition 24. *Let $n = n_1 n_2$, with $\gcd(n_1, n_2) = 1$, let g be a generator, let $\bar{\ell}_1, \bar{\ell}_2, \ldots, \bar{\ell}_l$ be elements of G, and let H be the subgroup of G of order n_1.*

Suppose that for a group element y chosen uniformly at random from G, we can find a relation of the form

$$y = \prod_{j=1}^{l} \bar{\ell}_j^{s_j}$$

with probability σ, using at most τ arithmetic operations.

Then for a group element y' chosen uniformly at random from H, we can find a relation of the form

$$y' = \prod_{j=1}^{l} (\bar{\ell}_j^{n_2})^{s'_j}$$

with probability σ, using at most $\tau + \chi + 2l + 1$ arithmetic operations, where χ is the number of arithmetic operations required for an exponentation in G.

Proof. If y' has been chosen uniformly at random from H, and b has been chosen uniformly at random from $\{0, 1, \ldots, n_2 - 1\}$, then

$$x = y' g^{n_1 b}$$

has been chosen uniformly at random from G.

By assumption, with probability β we can find $s_1, s_2, \ldots, s_l$ such that

$$x = \prod_{j=1}^{l} \bar{\ell}_j^{s_j}. \tag{10}$$

Let d be some inverse of n_2 modulo n_1. Then

$$y' = x^{n_2 d} = \prod_{j=1}^{l} \bar{\ell}_j^{s_j n_2 d} = \prod_{j=1}^{l} (\bar{\ell}_j^{n_2})^{s_j d}.$$

If we let s'_j be the remainder of $s_j d$ divided by n_1, we have a relation of the required form, and we succeed with probability β.

Generating x requires χ arithmetic operations. Finding (10) requires τ arithmetic operations. Computing d requires one arithmetic operation, and then computing $s'_1, s'_2, \ldots, s'_l$ requires $2l$ arithmetic operations. □

This result says that if we can find relations in the big group, we can move that relation into a subgroup. By Proposition 23 and Exercise 33, we can compute discrete logarithms in the subgroup. Theorem 8 now applies. (In fact, we can do even better by reusing relations that we find in the big group for each of the subgroups, and not finding new relations for each subgroup.)

Now we let $G = \mathbb{F}_p$. We want a way to find relations of the form (9) for randomly chosen group elements.

Let $\ell_1, \ell_2, \ldots, \ell_l$ be the l smallest primes (listed in order, so that $\ell_1 = 2$, $\ell_2 = 3$, etc.), and let $\bar{\ell}_1, \bar{\ell}_2, \ldots, \bar{\ell}_l$ be the corresponding group elements in $\mathbb{F}_p^*$. Define the sets

$$P = \left\{ \prod_{j=1}^{l} \ell_j^{s_j} < p \;\middle|\; s_1, s_2, \ldots, s_l \geq 0 \right\} \quad \text{and} \quad \bar{P} = \{k + \langle p \rangle \mid k \in P\}.$$

We have a natural bijection $\iota : \mathbb{F}_p \to \{0, 1, \ldots, p-1\}$ taking $k + \langle p \rangle$ to $k \bmod p$. This bijection obviously restricts to a bijection from $\bar{P}$ to P.

Next, we have an obvious map from P to $\mathbb{Z}^l$, taking $\prod \ell_j^{s_j}$ to $(s_1, s_2, \ldots, s_l)$. We extend this map to a map $\phi : \{1, 2, \ldots, p-1\} \to \mathbb{Z}^l \cup \{\perp\}$ by mapping any integer not in P to $\perp$.

Exercise 34. Show that we can compute the map ϕ using at most $\tau = \lfloor l + \log_2 p \rfloor$ integer divisions.

Exercise 35. Show that we can use the map ϕ and the algorithm implied by the previous exercise to generate relations of the form (9) with probability $\beta = |\bar{P}|/(p-1)$ using at most $\tau = \lfloor l + \log_2 p \rfloor$ arithmetic operations.

Exercise 36. A group operation in $\mathbb{F}_p^*$ requires two arithmetic operations (one multiplication and one division). Use Proposition 2 to show that we can compute an exponentiation (with an exponent smaller than the group order) in $\mathbb{F}_p^*$ using at most $4 \log_2 p$ arithmetic operations.

For simplicity, we consider a single large prime divisor q of $p-1$ such that q^2 does not divide $p-1$. With the above results, Proposition 23 and Exercises 34 and 36 say that we can compute logarithms modulo q using

$$\beta^{-1}(l+1)(l + \log_2 p + 8 \log_2 p) + (l+1)^3 + 2l + 3$$

arithmetic operations, where $\beta = |\bar{P}|/(p-1)$. We expect l to be much larger than $\log_2 p$, so we get an approximate cost

$$\beta^{-1} l^2 + l^3.$$

It is now clear that while we can make the fraction β large by making l large, that will increase the cost of generating relations and may actually increase the total cost. Making l too large may be counterproductive, and we need to find a good value.

We must estimate β. With $u = \log p / \log \ell_l$, we can crudely estimate that

$$\beta^{-1} = \frac{p-1}{|\bar{P}|} \approx u^u = \exp\left(\frac{\log p}{\log \ell_l} (\log\log p - \log\log \ell_l) \right).$$

By the prime number theorem

$$l \approx \ell_l / \log \ell_l = \exp\left(\log \ell_l - \log\log \ell_l \right),$$

giving us an approximate cost of

$$\exp\left(\frac{\log p \log\log p}{\log \ell_l} + 2 \log \ell_l - \left(2 + \frac{1}{\log \ell_l}\right) \log\log \ell_l \right) + \exp\left(3 \log \ell_l - 3 \log\log \ell_l \right).$$

Ignoring the $\log\log \ell_l$ terms, we want to choose ℓ_l so as to make the sum

$$\exp\left(\frac{\log p \log\log p}{\log \ell_l} + 2\log \ell_l\right) + \exp\left(3\log \ell_l\right)$$

as small as possible. The sum is dominated by the exponential with the biggest exponent. The second exponent increases monotonously with increasing ℓ_l. Since the first exponent has its minimum $\sqrt{8}\sqrt{\log p \log\log p}$ at

$$\log \ell_l = \sqrt{\log p \log\log p/2}$$

where the second exponent takes the smaller value $\sqrt{9/2}\sqrt{\log p \log\log p}$, we see that taking this value for $\log \ell_l$ should approximate a minimum. In particular, by using this value we should be able to compute discrete logarithms using approximately

$$\exp\left(\sqrt{8}\sqrt{\log p \log\log p}\right)$$

arithmetic operations.

We have arrived at the following requirement.

Requirement 6. If G is any subgroup of $\mathbb{F}_p^*$, then $\exp(\sqrt{8\log p \log\log p})$ arithmetic operations must be an infeasible computation.

Today, we have much better algorithms for computing discrete logarithms in finite fields. While we shall not study these algorithms, we note that the above requirement is not the final requirement.

4.4 Constructing Suitable Primes

5 Elliptic Curves

Elliptic curves have been studied for a long time by number theorists and a rich and varied theory has been developed. We are interested in elliptic curves because the points on an elliptic curve over a finite field forms a group that is suitable for use in cryptography.

From a mathematical point of view, studying elliptic curves over any field is interesting. From a cryptographic point of view, our groups come from elliptic curves over finite fields, which must therefore be our main interest. To simplify our presentation, we shall restrict ourselves to elliptic curves of a special form defined over prime fields. We note that essentially all of the theory we discuss works equally well for elliptic curves defined over other fields, though sometimes with minor modifications.

Even though we only discuss curves over finite prime fields, it is still convenient to use drawings of curves over the real numbers to illustrate ideas.

We begin by considering the algebraic curve C defined over the field $\mathbb{F}_p$, p a large prime, given by the polynomial equation

$$Y^2 = X^3 + AX + B, \qquad A, B \in \mathbb{F}_p. \tag{11}$$

Figure 2: Four cubic curves. From left to right: $Y^2 = X^3 - 6X - 1$, $Y^2 = X^3 - 2X + 2$, $Y^2 = X^3$ and $Y^2 = X^3 - \frac{3}{4}X + \frac{1}{4}$.

The *points* on the curve are the points in the plane that satisfy the curve equation. However, we cannot restrict the coordinates of the points to be elements of $\mathbb{F}_p$. We fix an algebraic closure $\bar{\mathbb{F}}$ of $\mathbb{F}_p$ and consider the points on the curve to be all the pairs $(x, y) \in \bar{\mathbb{F}}^2$ satisfying the curve equation.

A point on a curve is $\mathbb{F}_p$-*rational* (or just *rational*) if its coordinates lie in $\mathbb{F}_p$.

A curve is *smooth* if its partial derivatives never vanish all at the same time for points on the curve. Figure 2 shows three cubic curves, of which two are smooth.

The slope of a line passing through two distinct points on the curve is well-defined. Over a finite field, the tangent line no longer has a natural definition in terms of limits, but we can still use formal partial derivatives to define tangent lines.

Exercise 37. Show that a smooth curve has a well-defined, unique tangent line at any point on the curve.

Proposition 25. *An algebraic curve defined by* (11) *is smooth if and only if the polynomial $X^3 + AX + B$ has three distinct zeros.*

Proof. We first compute the partial derivative with respect to Y to get

$$2Y = 0.$$

Any point on the curve where both partial derivatives vanish must therefore have Y-coordinate 0.

It follows that the X-coordinate of a point where both partial derivatives vanish will be a zero of both $X^3 + AX + B$ and its derivative $3X^2 + A$. The only way a polynomial and its derivative can have a common zero is if the polynomial has a double or triple zero. We may conclude that the curve is smooth if and only if the polynomial $X^3 + AX + B$ has three distinct zeros. □

Fact 26. *A polynomial $aX^3 + bX^2 + cX + d$ has three distinct zeros if and only if its* discriminant *$b^2c^2 - 4ac^3 - 4b^3d - 27a^2d^2 + 18abcd$ is non-zero.*

In general, an elliptic curve is a smooth cubic curve, but we shall restrict attention to curves of the form we have already discussed.

Definition 5. An *elliptic curve* E over the field $\mathbb{F}_p$ is a cubic curve over $\mathbb{F}_p$ given by (11) with $4A^3 + 27B^2 \neq 0$.

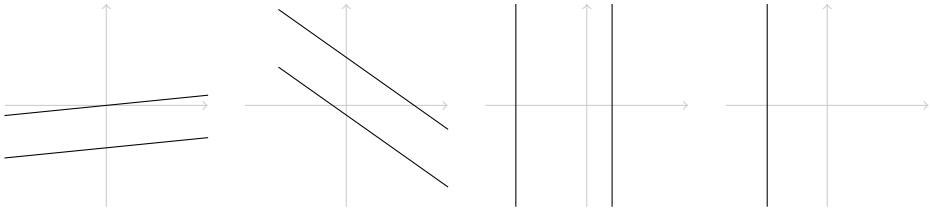


Figure 3: Intersection of lines and elliptic curves. From left to right: three distinct intersection points, possibly with complex Y-coordinates; tangent lines with one or two distinct intersection points; two distinct intersection points, possibly with complex Y-coordinates; and a tangent line with one intersection point.

We need to study the points of intersection between elliptic curves and straight lines. The situation can be neatly illustrated over the real numbers as in Figure 3. We see that lines can intersect the curve in zero, one, two or three points. This is untidy, and we want a better understanding of what is going on.

Proposition 27. *Let E be an elliptic curve defined by $Y^2 = X^3 + AX + B$.*

If L is a line defined by $Y = \alpha X + \beta$, then the zeros of the polynomial

$$X^3 - \alpha^2 X^2 + (A - 2\alpha\beta)X + B - \beta^2 \tag{12}$$

are the X-coordinates of the points of intersection.

If L is a line defined by $X = \beta$, then the zeros of the polynomial

$$Y^2 - \beta^3 - A\beta - B \tag{13}$$

are the Y-coordinates of the points of intersection.

Proof. We begin with a line L of the form $Y = \alpha X + \beta$. We want to compute the intersection of this line and an elliptic curve defined by $Y^2 = X^3 + AX + B$. Using the line equation to eliminate Y from the curve equation, we find

$$\alpha^2 X^2 + 2\alpha\beta X + \beta^2 = X^3 + AX + B.$$

The solutions to this equation corresponds to the zeros of the cubic polynomial (12). If x is any zero of this polynomial, we know that the point $(x, \alpha x + \beta)$ is on both the line and the curve and therefore a point of intersection.

Next we consider a line L' of the form $X = \beta$. This time, we use the line equation to eliminate X from the curve equation and get

$$Y^2 = \beta^3 + A\beta + B.$$

The solutions to this equation corresponds to the zeros of the quadratic polynomial (13). If y is any zero of this polynomial, we know that the point (β, y) is on both the line and the curve, and therefore a point of intersection. □

Fact 28. *Let $\mathbb{F}$ be a field and let $\bar{\mathbb{F}}$ be an algebraic closure of $\mathbb{F}$. Counting multiplicities, a polynomial of degree d over $\mathbb{F}$ has d zeros in $\bar{\mathbb{F}}$.*

Definition 6. Let E be an elliptic curve, P a point on E and L a line intersecting E in P. The *multiplicity* of the point of intersection P is the multiplicity of the corresponding zero of the polynomials in (12) or (13).

Exercise 38. Let E be an elliptic curve, P a point on E and L a line intersecting E in P. Show that the multiplicity of the intersection point is greater than 1 if and only if L is tangent to E at P.

Exercise 39. Let E be an elliptic curve. Show that E has 0, 1 or 3 rational points with vertical tangents.

Returning to Figure 3, in the left drawing we see two lines, one of which clearly has three real points of intersection. In this case, the polynomial (12) has three real solutions. The second line seemingly has only one point of intersection. In this case, (12) has only one real zero. But it also has two complex zeros, and these zeros correspond to points of intersection, points with complex coordinates. Over a general field, all three points of intersection may have X-coordinates in an extension field.

The second drawing in Figure 3 again has two lines, one of which has two points of intersection and one which has a single point of intersection. The corresponding cubic polynomial (12) does not have complex zeros, but double or triple zeros. The intersection point has multiplicity equal to the multiplicity of the corresponding zero. We do not have three distinct points of intersection, but if we count multiplicity, we have three points of intersection.

In the third drawing in Figure 3, by considering complex coordinates, we see that both lines have two distinct points of intersection. In the fourth drawing, by counting multiplicity, we get two points of intersection. But we never get three points of intersection, because (13) is a polynomial of degree two.

It is very inconvenient that vertical lines have only two points of intersection, while non-vertical lines have three points of intersection. It would have been nice if every line intersected the curve in three points, counting multiplicity. The proper solution to this issue lies in projective geometry, but in this text we shall choose a much simpler solution.

We declare that one extra point $\mathcal{O}$ exists with the following properties:

$\mathcal{O}$-1. The point $\mathcal{O}$ does not lie in the plane (hence has no coordinates), but lies on the curve and is a rational point.

$\mathcal{O}$-2. Any line of the form $X = \beta$ intersects the elliptic curve in $\mathcal{O}$ with multiplicity one. No line of the form $Y = \alpha X + \beta$ intersects the curve in $\mathcal{O}$.

$\mathcal{O}$-3. The curve has a tangent line in $\mathcal{O}$, and that line intersects the curve only in $\mathcal{O}$, with multiplicity three.

The special point $\mathcal{O}$ is often called the *point at infinity*.

As we saw in the discussion of Figure 3, considering coordinates in an algebraic closure and the notion of multiplicity somewhat simplifies the situation with regard

to intersection points. Now that we have introduced the point at infinity, we get the following nice results.

Proposition 29. *Let E be an elliptic curve defined over $\mathbb{F}_p$, and let L be a line. Counting multiplicities, L intersects E in exactly three points. If two of the intersection points are rational, then the third point is also rational.*

Proof. The first claim follows trivially from Proposition 27 and the properties of $\mathcal{O}$.

If the two points are both $\mathcal{O}$, then by $\mathcal{O}$-2 and $\mathcal{O}$-3 the line must be the tangent line to E at $\mathcal{O}$, which means that the third point of intersection is again $\mathcal{O}$, which is rational.

If only one point is $\mathcal{O}$, then by $\mathcal{O}$-2 the line takes the form $X = \beta$, and since it must pass through one point (x, y) where $x \in \mathbb{F}_p$, β must also lie in $\mathbb{F}_p$. It is clear from (13) that the third point of intersection will be $(x, -y)$, which is a rational point.

Finally, suppose neither of the two rational points is $\mathcal{O}$. We consider two cases. If the line is of the form $X = \beta$, then by $\mathcal{O}$-2 the third point of intersection is $\mathcal{O}$, which is rational.

Otherwise, the line is of the form $Y = \alpha X + \beta$. Since this line passes through two rational points, we know that $\alpha, \beta \in \mathbb{F}_p$. This means that the polynomial (12) has coefficients in $\mathbb{F}_p$. Furthermore, two of its zeros lie in $\mathbb{F}_p$, which means that the third zero also is in $\mathbb{F}_p$. This means that the X-coordinate of the third point of intersection lies in $\mathbb{F}_p$, and since it lies on the line $Y = \alpha X + \beta$, the third point of intersection is rational. □

Proposition 30. *Let P, Q be points (not necessarily distinct) on an elliptic curve E. Then there exists a unique line L and a unique point R such that P, Q and R are the points of intersection of E and L.*

Proof. If $P = Q = \mathcal{O}$ then $\mathcal{O}$-2 and $\mathcal{O}$-3 say that $R = \mathcal{O}$.

If P and Q are distinct points and one of them is $\mathcal{O}$, then the vertical line through the other point is determined by that point. By Proposition 29 this line intersects the curve in one more point, though not necessarily distinct.

If P and Q are distinct points and neither of them is $\mathcal{O}$, the line through the points is uniquely determined. By Proposition 29 this line intersects the curve in one more point, though not necessarily distinct.

If $P = Q \neq \mathcal{O}$, then the line we are looking for must be a tangent line. By Exercise 37 the tangent line is unique, and by Proposition 29 this line intersects the curve in one more point, though not necessarily distinct. □

We conclude this section by saying precisely what the points on an elliptic curve are.

Definition 7. The *points* on an elliptic curve E over $\mathbb{F}_p$ defined by

$$Y^2 = X^3 + AX + B, \qquad 4A^3 + 27B^2 \neq 0,$$

are the points with coordinates in the algebraic closure $\bar{\mathbb{F}}$ of $\mathbb{F}_p$ satisfying the curve equation, plus the special point $\mathcal{O}$:

$$E(\bar{\mathbb{F}}) = \{(x, y) \in \bar{\mathbb{F}}^2 \mid y^2 = x^3 + Ax + B\} \cup \{\mathcal{O}\}.$$

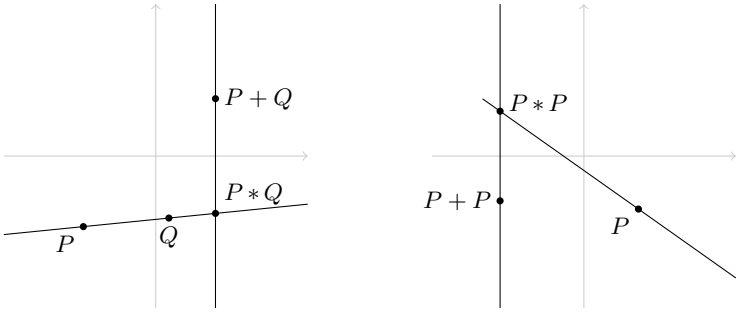


Figure 4: The group operation on elliptic curves. Addition of distinct points is shown on the left, point doubling is shown on the right.

The $\mathbb{F}_p$-*rational* (or just rational) points on E are the points with coordinates in $\mathbb{F}_p$, plus the special point $\mathcal{O}$:

$$E(\mathbb{F}_p) = \{(x, y) \in \mathbb{F}_p^2 \mid y^2 = x^3 + Ax + B\} \cup \{\mathcal{O}\}.$$

Since the rational points on an elliptic curve will form a group that we will use for cryptography, we are very interested in how many rational points there are on an elliptic curve.

Fact 31 (Hasse's theorem)**.** *Let E be an elliptic curve defined over $\mathbb{F}_p$. The number of rational points on E is*

$$|E(\mathbb{F}_p)| = p + 1 - t,$$

where $|t| \leq 2\sqrt{p}$.

5.1 Group Operation

We are now ready to turn the set of points on an elliptic curve into a group. We begin by defining a binary operation on the points of the elliptic curve, and then define the actual group operation in terms of the binary operation.

Definition 8. Let E be an elliptic curve. We define two binary operations $*$ and $+$ on as follows: $P * Q$ is the unique point identified by Proposition 30, and $P + Q = (P * Q) * \mathcal{O}$.

We shall show that there exists an identity element for $+$, there exists inverses for $+$ and that it is commutative. There are many ways to show that $+$ is associative, but they are either tedious or advanced, so we do not prove associativity.

Exercise 40. Let E be an elliptic curve and let P, Q be points on E. Show that $P * Q = Q * P$, and consequently that $P + Q = Q + P$.

Proposition 32. *Let E be an elliptic curve and let P be a point on E. Then $P + \mathcal{O} = \mathcal{O} + P = P$.*

Proof. First suppose $P = \mathcal{O}$. Then by Exercise 38 the line intersecting the curve in $\mathcal{O}$ with multiplicity at least 2 is a tangent, which by $\mathcal{O}$-3 intersects with multiplicity 3. It follows that $\mathcal{O} * \mathcal{O} = \mathcal{O}$ and that $\mathcal{O} + \mathcal{O} = \mathcal{O}$.

Next, suppose $P \neq \mathcal{O}$. By $\mathcal{O}$-3 the tangent through $\mathcal{O}$ and P intersects the curve in some point $Q \neq \mathcal{O}$ and $P * \mathcal{O} = Q$. It then follows that the line through P and Q has $\mathcal{O}$ as its third point of intersection, which means that $P + \mathcal{O} = P$. □

Proposition 33. *Let E be an elliptic curve and let $P = (x, y)$ be a point on E. Let $Q = (x, -y)$. Then Q is also on the curve, $P * \mathcal{O} = Q$ and $P + Q = \mathcal{O}$.*

Proof. It is immediately clear that if P is on the curve, then so is Q.

If $y = 0$, then $P = Q$ and the tangent in that point is a vertical line, which intersects the curve in $\mathcal{O}$ by $\mathcal{O}$-2, so $P * Q = \mathcal{O}$ and $P + Q = \mathcal{O}$.

If $y \neq 0$, the line through P and Q is vertical, so by $\mathcal{O}$-2 intersects the curve in $\mathcal{O}$. It follows that $P * Q = \mathcal{O}$, that $P + Q = \mathcal{O}$ and that $P * \mathcal{O} = Q$. □

Fact 34. *Let E be an elliptic curve and let P, Q, R be points on E. Then $(P+Q)+R = P + (Q + R)$.*

Theorem 35. *Let E be an elliptic curve. The set of points on E is a commutative group under $+$. The set of rational points is a subgroup.*

Proof. The fact that the set of points is a group follows from Propositions 32 and 33 and Fact 34. Commutativity follows from Exercise 40.

That the rational points form a subgroup follows from Proposition 30. □

We conclude this section by developing explicit formulae for computing $P + Q$.

By Proposition 32, if either point to be added is $\mathcal{O}$, the answer is the other point. Otherwise, we may assume that $P = (x_1, y_1)$ and $Q = (x_2, y_2)$.

By Proposition 33, if $x_1 = x_2$ and $y_1 = -y_2$, then the answer is $\mathcal{O}$.

If the answer has not been found, we need to find the third point of intersection of the line through P and Q. We begin by finding the slope α of the line. If $x_1 = x_2$, then $P = Q$ and we must use the tangent. The tangent line has slope

$$\alpha = \frac{3x_1^2 + A}{2y_1}.$$

If $x_1 \neq x_2$, then the slope is

$$\alpha = \frac{y_2 - y_1}{x_2 - x_1}.$$

In either case, the line's constant term is $\beta = y_1 - \alpha x_1$.

The X-coordinates of the intersection points of this line and the elliptic curve are zeros of (12). We know two of the zeros, namely x_1 and x_2, and we need to find the third zero x_3. The monic polynomial in (12) should be equal to the polynomial $(X - x_1)(X - x_2)(X - x_3)$. Comparing the coefficients of the X^2 term, we get that $-\alpha^2 = -x_1 - x_2 - x_3$, or

$$x_3 = \alpha^2 - x_1 - x_2.$$

The Y-coordinate of the third point of intersection is $\alpha x_3 + y_1 - \alpha x_1 = \alpha(x_3 - x_1) + y_1$. By Proposition 33, the Y-coordinate of $P + Q$ is

$$y_3 = \alpha(x_1 - x_3) - y_1.$$

We summarize the above in the following result.

Proposition 36. *Let E be an elliptic curve and P, Q be points on E.*

- *If $P = \mathcal{O}$, then $P + Q = Q$.*
- *If $Q = \mathcal{O}$, then $P + Q = P$.*

If neither point is $\mathcal{O}$, then let $P = (x_1, y_1)$ and $Q = (x_2, y_2)$.

- *If $x_1 = x_2$ and $y_1 = -y_2$, then $P + Q = \mathcal{O}$.*
- *Otherwise, $P + Q = (x_3, y_3)$ with*

$$x_3 = \alpha^2 - x_1 - x_2 \qquad \text{and} \qquad y_3 = \alpha(x_1 - x_3) - y_1,$$

where

$$\alpha = \begin{cases} \frac{3x_1^2 + A}{2y_1} & x_1 = x_2, \\ \frac{y_2 - y_1}{x_2 - x_1} & x_1 \neq x_2. \end{cases}$$

We conclude by noting that the *exponentiation* we have studied in Section 2, 3 and 4 now corresponds to a *point multiplication*

$$aP = \underbrace{P + P + \cdots + P}_{a \text{ terms}},$$

since the group of points on an elliptic curve is written additively instead of multiplicatively.

Even though the notation we have chosen for the elliptic curve group operation suggests addition and not multiplication, we use the words from Definition 3 and say that the discrete logarithm of a point Q to the base P is the smallest non-negative integer a such that $Q = aP$. We write $\log_P Q = a$.

Exercise 41. Redo Exercise 2 for point multiplications. Is anything different?

5.2 Point Counting

We have shown that the points on an elliptic curve form a commutative group, and the set of rational points is a finite commutative group. As we saw in Section 3, we need a cyclic group whose order is divisible by a large prime.

Exercise 42. Show that an elliptic curve has 0, 1 or 3 rational points of order 2.

Hint: Use Exercise 39.

Exercise 43. Let E be the curve defined by $Y^2 = X^3 - 13X + 12$ over the field $\mathbb{F}_{13}$. Show that E is not cyclic.

In general, the group of rational points is not cyclic, but it must contain a large cyclic subgroup.

Fact 37. *Let E be an elliptic curve defined over $\mathbb{F}_p$. Then there exists n_1, n_2, n_1 dividing both n_2 and $p-1$, such that*

$$E(\mathbb{F}_p) \simeq \mathbb{Z}^+_{n_1} \times \mathbb{Z}^+_{n_2}.$$

But a large cyclic subgroup is not sufficient for our purposes, we also need to know that its order is divisible by a large prime.

Proposition 38. *Let p be a prime congruent to 2 modulo 3, and let E be an elliptic curve defined by $Y^2 = X^3 + B$ over $\mathbb{F}_p$. Then $|E(\mathbb{F}_p)| = p + 1$.*

Proof. Since $p \equiv 2 \pmod 3$, 3 will be invertible modulo $p-1$. Then the map $\zeta \mapsto \zeta^3$ is invertible. If k is an inverse of 3 modulo $p-1$, then $\zeta \mapsto \zeta^k$ is the inverse map.

This means that for any value γ, the equation $X^3 + B$ has a unique solution in $\mathbb{F}_p$, and that solution is $(\gamma - B)^k$. We conclude that for every possible Y-coordinate y,

$$((y^2 - B)^k, y)$$

is a point on the curve. There are p such points and these are all the points with coordinates, which when counting $\mathcal{O}$ makes for $p+1$ points on the curve. □

Unfortunately, curves of the form $Y^2 = X^3 + B$ are so-called *supersingular*, which for reasons we shall not consider, are less suitable for our purposes than ordinary elliptic curves.

We shall briefly sketch Schoof's algorithm for counting points on an elliptic curve. The story begins with the *Frobenius map*. Let E be an elliptic curve defined over $\mathbb{F}_p$ and define a map $\phi : E(\bar{\mathbb{F}}) \to E(\bar{\mathbb{F}})$ by

$$\phi(P) = \begin{cases} \mathcal{O} & P = \mathcal{O}, \text{ and} \\ (x^p, y^p) & P = (x, y). \end{cases}$$

Exercise 44. Show that ϕ is a well-defined map.

Exercise 45. Show that ϕ is a group operation.

The fixed field of the map $\bar{\mathbb{F}} \to \bar{\mathbb{F}}$ given by $\alpha \mapsto \alpha^p$ is $\mathbb{F}_p$. It follows that the set of fixed points of ϕ is the set of rational points on E.

Hasse's theorem (Fact 31) says that there is a number t such that the number of rational points is $p + 1 - t$. It turns out that this number is closely related to the Frobenius map.

Fact 39. *Let E be an elliptic curve defined over $\mathbb{F}_p$, and let t be the integer such that the number of rational points on E is $p + 1 - t$. Then*

$$\phi^2(P) - t\phi(P) + pP = \mathcal{O} \tag{14}$$

for any point $P \in E(\bar{\mathbb{F}})$.

It turns out that the action of the Frobenius map on points of small order is important.

Definition 9. Let ℓ be an integer greater than 0. The set of ℓ-*torsion* points is

$$E[\ell] = \{P \in E(\bar{\mathbb{F}}) \mid \ell P = \mathcal{O}\}.$$

If $P \in E[\ell]$, then (14) becomes

$$\phi^2(P) - t_\ell \phi(P) + pP = \mathcal{O},$$

where $t_\ell = t \bmod \ell$. Reordering terms, we get

$$\phi^2(P) + pP = t_\ell \phi(P).$$

This is nothing more than a discrete logarithm problem, but note that the subgroup generated by $\phi(P)$ has order ℓ. If ℓ is not too big and computations on ℓ-torsion points are not too expensive, it will be possible to recover t_ℓ.

Note that $t \equiv t_\ell \pmod{\ell}$. If we can recover t_ℓ for many different, small primes ℓ whose product is greater than $4\sqrt{p}$, we can recover t and thereby the group order.

It still remains to show that we can find ℓ-torsion points and compute with them.

Fact 40. *Let E be an elliptic curve. There exists an efficiently computable sequence of polynomials (called* division polynomials*) $\psi_1(X,Y), \psi_2(X,Y), \ldots$ such that*

- *for any $P = (x,y) \in E(\bar{\mathbb{F}})$, $\psi_\ell(x,y) = 0$ if and only if $P \in E[\ell]$; and*
- *for odd ℓ, $\psi_\ell(X,Y)$ is a polynomial in X only, and has degree $\ell^2 - 1$.*

Fact 41. *Let E be an elliptic curve over $\mathbb{F}_p$. If p does not divide ℓ, then $E[\ell] \simeq \mathbb{Z}_\ell^+ \times \mathbb{Z}_\ell^+$.*

We now have a polynomial that characterizes all the X-coordinates of the ℓ-torsion points. Suppose that ℓ is an odd prime and smaller than p.

We can decide if $\psi_\ell(X)$ has rational zeros by computing $\gcd(X^p - X, \psi_\ell(X))$. It must either have 0, $(\ell-1)/2$ or $(\ell^2-1)/2$ rational zeros. Since each X-coordinate gives rise to two points, for the latter two cases we immediately know that $p + 1 - t$ is congruent to 0 modulo ℓ or ℓ^2, respectively.

More usually, $\psi_\ell(X)$ will not have any rational zeros. But it will usually not be irreducible. Let $f(X)$ be an irreducible factor of $\psi_\ell(X)$ of degree d. Then by constructing the extension field

$$\mathbb{F}_{p^d} \simeq \mathbb{F}_p[X]/\langle f(X) \rangle,$$

we know that the element $x \in \mathbb{F}_{p^d}$ corresponding to $X + \langle f(X) \rangle$ is a zero of $f(X)$ and hence of $\psi_\ell(X)$ and hence the X-coordinate of an ℓ-torsion point.

We now have the X-coordinate of an ℓ-torsion point, but we do not have a Y-coordinate. We can find a Y-coordinate by computing a square root of $x^3 + Ax + B$. Note that sometimes, we have to go to yet another field extension to find a square root, but this is unproblematic.

Factoring $\psi_\ell(X)$ is possible, but costly. Instead of finding an irreducible factor of $\psi_\ell(X)$, we can compute in the factor ring

$$\mathbb{F}_p[X]/\langle\psi_\ell(X)\rangle.$$

If $\psi_\ell(X)$ factors into irreducibles as $f_1(X)f_2(X)\cdots f_l(X)$, then

$$\mathbb{F}_p[X]/\langle\psi_\ell(X)\rangle \simeq \mathbb{F}_p[X]/\langle f_1(X)\rangle \times \cdots \times \mathbb{F}_p[X]/\langle f_l(X)\rangle.$$

Computing in this ring is just a simultaneous computation in every possible field extension where we could have found X-coordinates for ℓ-torsion points.

Since our computations involve divisions, and our ring contains non-invertible elements, the computation may not be possible to complete. However, when we cannot find an inverse of some element, we find a divisor of $\psi_\ell(X)$. If this happens, we simply restart the computation using one of the factors of $\psi_\ell(X)$. Eventually, we must either complete the computation or find an irreducible factor of $\psi_\ell(X)$.

Going into an extension of these rings to find the Y-coordinate of the ℓ-points is also unproblematic.

The above algorithm sketch works, but is very inefficient and mostly impractical. A more careful algorithm will work significantly faster.

The most important improvement to the algorithm is that for some small primes it is relatively easy to find an irreducible factor of $\psi_\ell(X)$. Since this factor has much smaller degree, we can work in a small field extension where arithmetic is much faster than in $\mathbb{F}_p[X]/\langle\psi_\ell(X)\rangle$. The resulting algorithm is significantly faster.

The upshot is that there are feasible algorithms that are able to compute the number of points on an elliptic curve. For a randomly chosen curve, the number of points is evenly distributed within the range given by Hasse's theorem, so we will not have to count the number of points of many curves until we find one with a suitable number of points.

5.3 Discrete Logarithms

In the group $\mathbb{F}_p^*$ the group operation requires two arithmetic operations (one integer multiplication and one integer division), while finding inverses is much more costly (using the extended Euclidian algorithm). Note that division in a finite field is usually done by multiplying with inverses.

In the group $E(\mathbb{F}_p)$, Proposition 36 says that adding distinct non-inverse points requires one inversion, three multiplications and six additions. Adding a point to itself requires one inversion, two multiplications by small constants, four multiplications, one addition of a constant and four additions.

At first glance, it would seem odd to consider the elliptic curve group, since the group operation there is much more complicated than the group operation in $\mathbb{F}_p^*$. But there is one more variable to consider: the size of the underlying field. Recall that we choose the size of the group such that the discrete logarithm problem in the group is sufficiently difficult.

The methods in Section 3 work for essentially any group. For $\mathbb{F}_p^*$ we also have index calculus methods, which become significantly better than the methods in Section 3 as p grows.

For most elliptic curves, there are equivalents of the small primes, so there are no useful index calculus methods.

For so-called *anomalous elliptic* curves, curves defined over $\mathbb{F}_p$ with p elements, there are very efficient algorithms for computing discrete logarithms. These curves are completely unsuitable for use in cryptography.

For certain subgroups G, H of $E(\overline{\mathbb{F}})$, there are so-called bilinear maps $e : G \times H \to \mathbb{F}_{p^d}^*$ satisfying

$$e(aP, bQ) = e(P, Q)^{ab}.$$

When the field extension degree d is small, these maps can be computed easily and can sometimes be used to move a discrete logarithm problem from an elliptic curve into a finite field.

Supersingular curves have very low extension degree, which means that the discrete logarithm problem on supersingular curves is not much harder than the discrete logarithm problem in certain finite fields. Since index calculus methods can be used in finite fields, supersingular curves must be defined over large finite fields, making arithmetic much slower.

Note that bilinear maps can sometimes be used constructively in cryptography. But unless we need easily computable bilinear maps, supersingular curves are not useful for cryptography.

For many other small families of elliptic curves, there are algorithms capable of computing discrete logarithms faster than the methods from Section 3. But for most elliptic curves, the best algorithms for computing discrete logarithms are those from Section 3. Compared to $\mathbb{F}_p^*$, our elliptic curves can therefore be defined over much smaller fields where arithmetic is much faster, so even if we have to do more arithmetic operations, each operation is faster.

6 Active Attacks

The only attackers we have considered so far are eavesdroppers. For some communications channels, this is correct. But for most channels in use today, an attacker that can eavesdrop can also tamper with communications.

In practice, Diffie-Hellman on its own will not be secure. We shall consider how Diffie-Hellman can be secured later.