Chapter 2

Discrete logarithms

In the setting of the RSA cryptosystem, once the keys have been set up and we have a public key (N, e), the two main processes are

(a) **encryption**: given $m < N$, compute $c \equiv m^e \pmod{N}$;

(b) **decryption**: given $c < N$, compute m such that $c \equiv m^e \pmod{N}$.

The first operation, raising to the power e, can be performed efficiently. The second operation is its inverse, taking e-th roots, and appears to be much more difficult if we only know N and e (and becomes easy again if we know d).

The map $(m, e) \mapsto m^e \pmod{N}$ is a function of two variables; in the RSA context we fix the variable e and let m vary.

But we could instead fix m and let e vary, getting functions

(a) **exponential in base** m: $\mathbf{Z} \longrightarrow (\mathbf{Z}/N\mathbf{Z})^\times$ given by $e \mapsto m^e \pmod{N}$;

(b) **logarithm in base** m: $(\mathbf{Z}/N\mathbf{Z})^\times \longrightarrow \mathbf{Z}$ given by $c \mapsto e$ with the property that $c \equiv m^e \pmod{N}$.

We define this in a more general setting: given an abelian group G and two elements $g, h \in G$, a *discrete logarithm* of h in base g is the smallest integer $e \in \mathbf{Z}_{\geqslant 1}$ such that $h = g^e$. (Note that this exists if and only if $h \in \langle g \rangle$. Note also that we are using multiplicative notation for the group, mostly to match the typical setting for logarithms, as well as one of the main examples, the group $(\mathbf{Z}/p\mathbf{Z})^\times$.)

The *discrete logarithm problem (DLP)* is then: find a discrete logarithm or show that it does not exist.

In certain groups, this problem is easily solved:

Example 2.1. Let $G = \mathbf{Z}/N\mathbf{Z}$ with addition. Let $g, h \in G$. Adapting to the additive notation, we need to solve for e in the congruence $eg \equiv h \pmod{N}$.

In other words, we want to find e such that there exists k with $eg - h = kN$. Rewriting this as $eg - kN = h$, we see that this is solvable if and only if $\gcd(g, N) \mid h$, and that e can be found by applying the extended Euclidean algorithm to g and N.

In particular, as you know from Assignment 1, this requires a number of steps that is linear in the size of g.

In other groups, the DLP is significantly harder, as we will soon see. For cryptographic purposes, these are the types of groups we are interested in.

2.1 Cryptographic applications of the DLP

We describe a couple of schemes in the general context of an abelian group G, keeping in mind that (a) for a practical application we have to find an appropriate group to work in and (b) the safety and efficiency properties of the practical schemes may depend heavily on the group we choose.

If you prefer a more concrete approach, feel free to substitute $G = (\mathbf{Z}/p\mathbf{Z})^\times$ for a prime p in the next couple of sections. (This is where the classical versions of these schemes live anyway.)

Diffie–Hellman key exchange

The cryptographic applications we have seen so far had as goal the secure transmission of a particular message.

We now consider a slightly different challenge: Alice and Bob want to generate a random key that they will then both have access to, but any information that they share in order to achieve this goal is passed along insecure channels, hence subject to eavesdropping. Can this be done in such a way that only Alice and Bob end up knowing the secret key? A mechanism for this task was devised by Diffie and Hellman in 1976.

The start is the choice of an abelian group G (ideally one where the DLP is hard) and of an element $g \in G$ (ideally such that g has large prime order in G). The pair (G, g) is made public; it could for instance be a standard pair chosen by a trusted entity.

Next Alice chooses a secret integer a and computes $A = g^a$; similarly, Bob chooses a secret integer b and computes $B = g^b$. They exchange A and B on the insecure communication channel.

Finally, Alice computes $B^a = g^{ab}$; Bob does the same and gets $A^b = g^{ab}$. Since $B^a = A^b$, we call this value K; it is the secret key that now both Alice and Bob have and can put to good use, for instance as a one-time pad.

Let's put ourselves in the place of the eavesdropper Eve. She knows G, g, A, and B, but not a or b; she wants to find the secret key K. In other words, she has to solve the *Diffie–Hellman problem*: given G, g, g^a, and g^b, compute g^{ab}.

If Eve has access to an easy solution to the DLP in G, then she is done. Solve for a in $g^a = A$ and compute B^a (or solve for b in $g^b = B$ and compute A^b).

In the other direction, it is not known whether solving the Diffie–Hellman problem for G allows one to solve the DLP for G.

ElGamal cryptosystem

In 1985, ElGamal developed the Diffie–Hellman approach into a full cryptosystem, where specific messages can be communicated (rather than just random shared keys).

Let's assume that Alice wants to send a message to Bob.

The initial setup is the same: an abelian group G where DLP is (presumed) hard and an element $g \in G$ of large prime order.

He then picks a secret integer b as private key and computes $B = g^b$. He sends B to Alice along the insecure channel.

Alice is looking to send a plaintext message $m \in G$ to Bob. Alice chooses a random integer a and computes the two quantities

$$A_1 = g^a, \qquad A_2 = mB^a.$$

The ciphertext is the pair $c = (A_1, A_2)$, which Alice sends to Bob; she then discards a, which should not be reused.

Bob's decryption is the computation of

$$A_2(A_1^b)^{-1} = (mB^a)(g^{ab})^{-1} = (mg^{ba})(g^{ab})^{-1} = m,$$

so that Bob now holds the plaintext message m.

Putting ourselves in Eve's position, we can state the *ElGamal problem*: given G, g, g^b, g^a, and mg^{ab}, compute m.

We will see that the ElGamal problem is at least as hard as the Diffie–Hellman problem.

2.2 Generic attacks on the DLP

We describe some algorithms for solving the discrete logarithm problem in an arbitrary abelian group G.

Recall that we are given the group G and two elements $g, h \in G$. We want to find the smallest positive integer e such that $h = g^e$.

We will assume that g has finite order N in G.

We are going to express the complexity of an algorithm in terms of basic operations in the group: multiplication $(x, y) \mapsto xy$, inverse $x \mapsto x^{-1}$, and comparison (given $x, y \in G$, determine whether $x = y$).

Brute force search

This is the most straightforward approach: let $x_1 = g$ and compare x_1 to h. If they are equal, return $e = 1$; otherwise let $x_2 = x_1 g$ and repeat. At each step we have $x_n = x_{n-1}g = g^n$ and we compare x_n and h.

If we reach x_{N-1} and the comparison did not succeed, then h is not in the cyclic subgroup $\langle g \rangle$ and the discrete logarithm $\log_g(h)$ does not exist. This takes at most $N - 2$ group operations (multiplications, in this case) and N comparisons.

The Pohlig–Hellman method

The idea is to use brute force search if N is prime, and otherwise to exploit the prime factorisation of N to reduce to the prime case.

A group-theoretic reframing of the DLP

The element g of order N defines a group isomorphism $\varphi : \langle g \rangle \longrightarrow \mathbf{Z}/N\mathbf{Z}$ where $\varphi(g) = 1$. If $h \in \langle g \rangle$, solving the DLP $h = g^e$ is equivalent to determining $\varphi(h) = e$. (If it helps, think "$\phi = \log_g$"; we'll be using that notation soon in a more specific setting.)

Consider the prime factorisation $N = \ell_1^{s_1} \dots \ell_r^{s_r}$. By the Chinese Remainder Theorem, we have a group isomorphism

$$\mathbf{Z}/N\mathbf{Z} \longrightarrow \mathbf{Z}/\ell_1^{s_1}\mathbf{Z} \times \cdots \times \mathbf{Z}/\ell_r^{s_r}\mathbf{Z}.$$

Fixing $j \in \{1, \dots, r\}$, we have a surjective group homomorphism

$$\mathbf{Z}/N\mathbf{Z} \longrightarrow \mathbf{Z}/\ell_j^{s_j}\mathbf{Z}, \qquad x \mapsto x \pmod{\ell_j^{s_j}}.$$

On the other hand, letting $N_j = N/(\ell_j^{s_j})$, we have a group homomorphism $f_j : \langle g \rangle \longrightarrow \langle g \rangle$ given by $f_j(g) = g^{N_j}$. It is easy to see that g^{N_j} has order $\ell_j^{s_j}$, and that the image of f_j is precisely $\langle g^{N_j} \rangle$. We

summarise the situation in a commutative diagram

$$\begin{array}{ccc} \langle g \rangle & \xrightarrow{\varphi} & \mathbf{Z}/N\mathbf{Z} \\ {\scriptstyle f_j}\downarrow & & \downarrow{\scriptstyle \pmod{\ell_j^{s_j}}} \\ \langle g^{N_j} \rangle & \xrightarrow[\varphi_j]{} & \mathbf{Z}/\ell_j^{s_j}\mathbf{Z}. \end{array}$$

Now let's chase the element $h \in \langle g \rangle$ around this diagram. We have $\varphi(h) = e$, so

$$(\varphi_j \circ f_j)(h) = e \pmod{\ell_j^{s_j}}.$$

In other words, we can get $e_j \equiv e \pmod{\ell_j^{s_j}}$ by solving the DLP $h^{N_j} = \left(g^{N_j}\right)^{e_j}$, where the order of the generator is $\ell_j^{s_j}$. Once all the e_j are found, we get the desired e by applying the Chinese Remainder Theorem to the system of congruences

$$\begin{aligned} e &\equiv e_1 \pmod{\ell_1^{s_1}}, \\ &\vdots \\ e &\equiv e_r \pmod{\ell_r^{s_r}}. \end{aligned}$$

The prime power case $N = \ell^s$

If $s = 1$ so $N = \ell$ is prime, we just use brute force search.

Let's consider the case where $N = \ell^2$ for some prime ℓ. Given g of order ℓ^2 and h, want to find e such that $h = g^e$, or determine that such e does not exist.

Write e in base ℓ: $e = e_0 + e_1\ell$, where the unknown e_0, e_1 satisfy $0 \leqslant e_i < \ell$.

Suppose $h = g^e$. Let $g_1 = g^\ell$, then g_1 has order ℓ in G, and

$$h^\ell = g^{e\ell} = g_1^e = g_1^{e_0 + e_1\ell} = g_1^{e_0}.$$

We can then get e_0 by solving the DLP with base g_1 of order ℓ.

To get the other digit e_1, let $h_1 = hg^{-e_0}$, then $h_1 = g^{e_1\ell} = g_1^{e_1}$. So we can get e_1 by solving another DLP with base g_1.

The longest this will take is if $e_0 = \ell - 1$ and e_1 does not exist (that is, h_1 is not in the cyclic subgroup generated by g_1). Tallying the necessary group operations:

- computing $g_1 = g^\ell$ takes at most $2\log_2(\ell)$ multiplications;
- computing h^ℓ takes at most $2\log_2(\ell)$ multiplications;
- finding e_0 takes at most $\ell - 2$ multiplications by brute force;
- computing $h_1 = h(g^{e_0})^{-1}$ takes at most $2\log_2(\ell - 1) + 1$ multiplications and one inversion;
- finding e_1 takes at most $\ell - 2$ multiplications by brute force;

All in all we have at most $2\ell + 6\log_2(\ell) - 2$ group operations.

Comparing this to the $\ell^2 - 2$ operations necessary for an order ℓ^2 brute force search, we definitely win for large ℓ.

The approach can be extended in a fairly straightforward way to arbitrary powers $N = \ell^s$.

The overall complexity of the Pohlig–Hellman method is given by

Theorem 2.2. *If g has order $N = \ell_1^{s_1} \dots \ell_r^{s_r}$ and $B = \max\{\ell_1, \dots, \ell_r\}$ then the Pohlig–Hellman method solves the DLP in $O\left(\log(N)^2 + B\log(N)\right)$ group operations.*

We can look at this in two ways:

- The direct dependence on N is polynomial in the size $\log(N)$, so to get a "hard" DLP we need B to be large; the best we can do is N prime, in which case Pohlig–Hellman is simply brute force search.
- The direct dependence on N is polynomial in the size $\log(N)$; to lower the overall time complexity we could try to do better than brute force in the prime case.

Baby-step giant-step algorithm

The idea is to store a large number of auxiliary values that can then be used to solve the DLP. It is a situation where we trade space (computer memory) for a speedup in time.

Suppose g has order N, which is a (large) prime, and $h = g^e$ for some e with $0 \leqslant e < N$. Setting $m = \left\lceil \sqrt{N} \right\rceil$, we can write e in base m:

$$e = e_0 + e_1 m, \qquad 0 \leqslant e_0, e_1 < m.$$

Then

$$h = g^e = g^{e_0}(g^m)^{e_1} \quad \Rightarrow \quad g^{e_0} = h(g^{-m})^{e_1}.$$

With this in mind, we compute the set $L = \{g^{e_0} : \ 0 \leqslant e_0 < m\}$ (the *baby steps*) of all possible values for the LHS of the equation.

Then we do the *giant steps*: start at $e_1 = 0$, compute $x = h(g^{-m})^{e_1}$, and look for $x \in L$. If we find it, then we have our e_0 and e_1, and therefore $e = e_0 + e_1 m$. If we do not find x, we increment e_1 and try again. If we get to $e_1 = m$, then $h \notin \langle g \rangle$ so the DLP is not solvable.

Example 2.3. Take $G = (\mathbf{Z}/83\mathbf{Z})^\times$, $g = 4$. We have $\phi(83) = 82 = 2 \cdot 41$ and 2 is a primitive root of 83, so $g = 2^2$ has order $N = 41$.

We get $m = \left\lceil \sqrt{41} \right\rceil = 7$, and the baby steps table is

e_0	0	1	2	3	4	5	6
g^{e_0}	1	4	16	64	7	28	29

One further multiplication gives us $g^m = g^7 = 33$, then the extended Euclidean algorithm gives $g^{-m} = 78$.

We are ready for the giant steps. Let's try $h = 9$:

e_1	0	1	2	3	4
$h(g^{-m})^{e_1}$	9	38	59	37	64

We see that 64 is in the baby steps table, so we are done: $e_1 = 4$, $e_0 = 3$, so $e = 3 + 4 \cdot 7 = 31$.

It is easy to check that indeed $4^{31} \equiv 9 \pmod{83}$.

If we try $h = 5$ instead, we get giant steps

e_1	0	1	2	3	4	5	6
$h(g^{-m})^{e_1}$	5	58	42	39	54	62	22

Having gotten to this point without collisions, we conclude that $5 \notin \langle 4 \rangle$, which is indeed the case.

The baby steps require $m-2$ group operations and as many storage operations into L. In the worst (unsolvable) case the giant steps require about m group operations, and as many lookups in L. Depending on how clever the storage scheme for L is, storage/lookup can add some overhead; with the most efficient scheme (a hash table) there is a time complexity of about $O(m) = O(\sqrt{N})$, significantly better than the $O(N)$ of brute force search.

The space needed by the algorithm is roughly the size of baby steps set L, basically $O(m) = O(\sqrt{N})$. If this is prohibitive, one can take $m = \lceil N^{1/3} \rceil$ and write $e = e_0 + e_1 m + e_2 m^2$, then use the identity

$$g^{e_0} = h(g^{-m})^{e_1}(g^{-m^2})^{e_2}.$$

The space and time needed for the baby steps is now $O(N^{1/3})$; the time needed for the giant steps can go up to $O(N^{2/3})$.

2.3 Non-linear congruences

Solving congruences using primitive roots

We specialise to the setting $G = (\mathbf{Z}/m\mathbf{Z})^\times = \langle g \rangle$, in other words g is a primitive root modulo m.

Recall from our group-theoretic discussion of the DLP that we have a group isomorphism $\log_g = \varphi : (\mathbf{Z}/m\mathbf{Z})^\times \longrightarrow \mathbf{Z}/\phi(m)\mathbf{Z}$.

This says that $\log_g(1) = 0$, $\log_g(g) = 1$, and importantly that

$$\log_g(ab) = \log_g(a) + \log_g(b),$$

which also implies that $\log_g(a^k) = k \log_g(a)$.

These identities involve elements of $(\mathbf{Z}/m\mathbf{Z})^\times$ as the arguments of the function $\log_g$, and elements of $\mathbf{Z}/\phi(m)\mathbf{Z}$ as the values of the function $\log_g$, so when we are dealing more generally with integers we express these identities as congruences with respect to the relevant modulus.

One advantage of having a group isomorphism is that it allows us to translate equations taking place in one group into equations in the other group. For instance, if we have a congruence $A \equiv B \pmod m$ and A, B are both coprime to m, then we can apply the logarithm and get the congruence $\log_g(A) \equiv \log_g(B) \pmod{\phi(m)}$. This is similar to the logarithm of real numbers, and just as useful.

Example 2.4. Solve the congruence $6x^5 \equiv 7 \pmod{17}$.

To start with, the inverse of 6 is 3, so we can rewrite the congruence as $x^5 \equiv 21 \equiv 4 \pmod{17}$.

Here $m = 17$. A primitive root modulo 17 is $g = 3$, so we take $\log_3$ on both sides of the last congruence:

$$5 \log_3(x) \equiv \log_3(4) \pmod{\phi(17)}.$$

and since $5^{-1} \equiv 13 \pmod{16}$, we end up with

$$\log_3(x) \equiv 13 \log_3(4) \equiv \log_3(4^{13}) \pmod{16}.$$

Now we apply the inverse homomorphism to $\log_3$:

$$x \equiv 4^{13} \pmod{17},$$

and a simple calculation gives $x \equiv 4 \pmod{17}$.

Example 2.5. Solve the congruence $8^x \equiv 2 \pmod{17}$.

Again $m = 17$ so we can use the primitive root $g = 3$. Take logarithms:

$$x \log_3(8) \equiv \log_3(2) \pmod{16}.$$

We can rewrite this as

$$(3x - 1) \log_3(2) \equiv 0 \pmod{16}.$$

Now $\log_3(2) = 14$, so we get

$$14(3x - 1) \equiv 0 \pmod{16}.$$

Since 2 divides both sides of the congruence as well as the modulus 16, we can remove a factor of 2 all across to get

$$7(3x - 1) \equiv 0 \pmod{8}.$$

At this point 7 is coprime to 8 so we can multiply by its inverse and get rid of it:

$$3x - 1 \equiv 0 \pmod{8} \Rightarrow 3x \equiv 1 \pmod{8} \Rightarrow x \equiv 3^{-1} \equiv 3 \pmod{8}.$$

So the solutions of the original equation are all integers that are $3 \pmod{8}$.

Quadratic (non-)residues

We consider the simplest type of non-linear congruence: $x^2 \equiv a \pmod{p}$. Since not much is happening modulo 2, we'll assume that p is odd.

Given an odd prime number p and $a \in (\mathbf{Z}/p\mathbf{Z})^\times$, when does the above congruence have solutions?

For instance, when $p = 7$, we have $1^2 \equiv 6^2 \equiv 1$, $2^2 \equiv 5^2 \equiv 4$, and $3^2 \equiv 4^2 \equiv 2$. Therefore, the congruence $x^2 \equiv a \pmod{7}$ has solutions for $a = 1, 2, 4$, and does not have solutions for $a = 3, 5, 6$.

We say that $a \in (\mathbf{Z}/p\mathbf{Z})^\times$ is a *quadratic residue modulo p* if $x^2 \equiv a \pmod{p}$ is solvable; otherwise a is a *quadratic non-residue modulo p*.

We'll be interested in theoretical aspects of this notion, as well as practical questions.

Proposition 2.6. *Let g be a primitive root modulo an odd prime p. Then $a \in (\mathbf{Z}/p\mathbf{Z})^\times$ is a quadratic residue if and only if $a = g^e$ with e even.*

Proof. In one direction, suppose $e = 2k$ and let $x = g^k$. Then

$$x^2 \equiv (g^k)^2 \equiv g^{2k} \equiv g^e \equiv a \pmod{p}.$$

For the other direction, we proceed by contradiction. Suppose $a = g^e$ with $e = 2k + 1$ and a is a quadratic residue, say $x^2 \equiv a \pmod{p}$. We have

$$1 \equiv x^{p-1} \equiv (x^2)^{(p-1)/2} \equiv a^{(p-1)/2} \equiv g^{k(p-1)} \cdot g^{(p-1)/2} \equiv g^{(p-1)/2} \pmod{p},$$

where we used Fermat's little Theorem twice. However, $g^{(p-1)/2} \equiv 1 \pmod{p}$ contradicts the fact that the primitive root g has order $p - 1$. □

Corollary 2.7. *Let p be an odd prime. The product of two quadratic residues is a quadratic residue. The product of two quadratic non-residues is a quadratic residue. The product of a quadratic residue and a quadratic non-residue is a quadratic non-residue.*

Corollary 2.8. *Let p be an odd prime. The subset R of quadratic residues is a subgroup of $(\mathbf{Z}/p\mathbf{Z})^\times$.*

In particular, $(\#R) \mid (p-1)$. In fact, $\#R = (p-1)/2$, as you will see in next week's tutorial.

The *Legendre symbol* is the function $\mathbf{Z} \longrightarrow \{-1, 0, 1\}$ given by the rule

$$\left(\frac{a}{p}\right) = \begin{cases} 1 & \text{if } a \pmod p \text{ is a quadratic residue,} \\ -1 & \text{if } a \pmod p \text{ is a quadratic non-residue,} \\ 0 & \text{if } p \mid a. \end{cases}$$

Corollary 2.9. *The Legendre symbol is multiplicative:*

$$\left(\frac{ab}{p}\right) = \left(\frac{a}{p}\right)\left(\frac{b}{p}\right) \qquad \textit{for all } a, b \in \mathbf{Z}.$$

Theorem 2.10 (Quadratic Reciprocity)**.** *If p and q are odd primes, then*

$$\left(\frac{p}{q}\right)\left(\frac{q}{p}\right) = (-1)^{\frac{p-1}{2}\frac{q-1}{2}}$$
$$\left(\frac{-1}{p}\right) = (-1)^{\frac{p-1}{2}}$$
$$\left(\frac{2}{p}\right) = (-1)^{\frac{p^2-1}{8}}$$

This was conjectured by Euler and Legendre and proved by Gauss, who was apparently really proud of it. Why? Here's one of the reasons:

Example 2.11. Is 21 a quadratic residue modulo 89?

We have by multiplicativity

$$\left(\frac{21}{89}\right) = \left(\frac{3}{89}\right)\left(\frac{7}{89}\right).$$

By Quadratic Reciprocity:

$$\left(\frac{3}{89}\right)\left(\frac{89}{3}\right) = 1,$$

so we have

$$\left(\frac{3}{89}\right) = \left(\frac{89}{3}\right) = \left(\frac{2}{3}\right) = -1.$$

Also by Quadratic Reciprocity:

$$\left(\frac{7}{89}\right)\left(\frac{89}{7}\right) = 1,$$

so we have

$$\left(\frac{7}{89}\right) = \left(\frac{89}{7}\right) = \left(\frac{5}{7}\right).$$

Applying Quadratic Reciprocity again, we have

$$\left(\frac{5}{7}\right) = \left(\frac{7}{5}\right) = \left(\frac{2}{5}\right) = (-1)^{(5^2-1)/8} = -1.$$

Putting it all together we conclude that $\left(\frac{21}{89}\right) = 1$, so 21 is a quadratic residue modulo 89.

If you don't believe a word of this: $33^2 = 1089 \equiv 21 \pmod{89}$.

If you think about this last example a bit, you will notice that at the very start we had to factor 21, which (thankfully) was not too challenging. But, in general, do we need to factor possibly large integers in order to use this method?

It turns out that the answer is no. The solution is to generalise the Legendre symbol and replace the prime p with an arbitrary odd positive integer b.

The *Jacobi symbol* is the function $\mathbf{Z} \longrightarrow \{-1, 0, 1\}$ given by the following rule: if $b = p_1^{e_1} \dots p_r^{e_r}$ then let

$$\left(\frac{a}{b}\right) = \left(\frac{a}{p_1}\right)^{e_1} \dots \left(\frac{a}{p_r}\right)^{e_r}.$$

Proposition 2.12. *The Jacobi symbol is multiplicative in two ways:*

$$\left(\frac{a_1 a_2}{b}\right) = \left(\frac{a_1}{b}\right)\left(\frac{a_2}{b}\right),$$
$$\left(\frac{a}{b_1 b_2}\right) = \left(\frac{a}{b_1}\right)\left(\frac{a}{b_2}\right).$$

Moreover:

$$\text{if } a_1 \equiv a_2 \pmod{b}, \text{ then } \left(\frac{a_1}{b}\right) = \left(\frac{a_2}{b}\right).$$

Even more interestingly, Quadratic Reciprocity extends nicely to this setting:

Theorem 2.13. *If a and b are odd positive integers then*

$$\left(\frac{a}{b}\right)\left(\frac{b}{a}\right) = (-1)^{\frac{a-1}{2}\frac{b-1}{2}}$$
$$\left(\frac{-1}{b}\right) = (-1)^{\frac{b-1}{2}}$$
$$\left(\frac{2}{b}\right) = (-1)^{\frac{b^2-1}{8}}$$

We can now redo the last example by a simple chain of applications of this theorem:

$$\left(\frac{21}{89}\right) = \left(\frac{89}{21}\right) = \left(\frac{5}{21}\right) = \left(\frac{21}{5}\right) = \left(\frac{1}{5}\right) = 1.$$

Quadratic residues and probabilistic encryption

We have just seen that the Jacobi symbol behaves a lot like the Legendre symbol. But there's a subtle difference that, as it turns out, can be exploited in an interesting way.

By the definition of the Legendre symbol, for any odd prime p we have that $\left(\frac{b}{p}\right) = 1$ if and only if b is a quadratic residue modulo p.

But the Jacobi symbol is not defined directly in terms of quadratic residues, so is there a relation?

Example 2.14. Consider $N = 143 = 11 \cdot 13$.

(a) Take $b = 3$. The Jacobi symbol is

$$\left(\frac{3}{143}\right) = -\left(\frac{143}{3}\right) = -\left(\frac{2}{3}\right) = 1.$$

Is 3 a quadratic residue modulo 143?

Well:

$$\left(\frac{3}{11}\right) = -\left(\frac{11}{3}\right) = -\left(\frac{2}{3}\right) = 1,$$

so by the definition of the Legendre symbol, 3 is a quadratic residue modulo 11. In fact, $5^2 \equiv 3 \pmod{11}$.

Similarly:

$$\left(\frac{3}{13}\right) = \left(\frac{13}{3}\right) = \left(\frac{1}{3}\right) = 1,$$

so 3 is a quadratic residue modulo 13. In fact, $4^2 \equiv 3 \pmod{13}$.

Since $\gcd(11, 13) = 1$, the Chinese Remainder Theorem provides a solution to the system of congruences

$$\begin{aligned} x &\equiv 5 \pmod{11}, \\ x &\equiv 4 \pmod{13}, \end{aligned}$$

namely $x = 4 \cdot 6 \cdot 11 - 5 \cdot 5 \cdot 13 = -61 \equiv 82 \pmod{143}$. We can then check that $82^2 \equiv 3 \pmod{143}$, so 3 is indeed a quadratic residue modulo 143.

(b) Take $b = 7$. The Jacobi symbol is

$$\left(\frac{7}{143}\right) = -\left(\frac{143}{7}\right) = -\left(\frac{3}{7}\right) = \left(\frac{7}{3}\right) = \left(\frac{1}{3}\right) = 1.$$

But

$$\left(\frac{7}{11}\right) = -\left(\frac{11}{7}\right) = -\left(\frac{4}{7}\right) = -1,$$

so by the definition of the Legendre symbol, 7 is not a quadratic residue modulo 11.

We conclude that 7 cannot be a quadratic residue modulo 143.

The moral of the story is that despite being easily determined, the Jacobi symbol does not in general know whether b is a quadratic residue.

We are back in a situation where, given $N = pq$, we have more information if we are given access to the factors p and q than if we are not. This should ring a bell. Maybe it sounds a bit like what follows.

Imagine a situation in which Alice wants to send Bob a message using a public key, and this message can only be one of two things, say 0 or 1. This is laughably foolish. After all, Eve is always listening. If she is aware that the only two possibilities are 0 and 1, she can use Bob's public key to encrypt both 0 and 1 separately, then compare her result to the ciphertext sent by Alice. Once she matches it, she knows what the message was.

This problem occurs whenever the space of possible plaintext messages is small.

The idea of *probabilistic encryption* is to artificially make the space much bigger by appending a random string r to the plaintext message m, then encrypting the pair (m, r). To an outside eavesdropper, the space of ciphertexts corresponding to $(0, r)$ (as r varies) and the space of ciphertexts corresponding to $(1, r)$ (as r varies) should look pretty much identical.

The trick is to organise this in such a way that Bob can retrieve the original m upon decryption (whether he gets the correct random part r or not is irrelevant).

A simple yet elegant way of achieving these goals was devised in 1982 by Goldwasser and Micali[1], based precisely on the Jacobi symbol quirk we noted in the previous example.

Here is how it goes. Bob picks random secret primes p and q and computes $N = pq$. He also chooses an integer b that is a quadratic non-residue modulo p and modulo q. He keeps one of the primes (say p) as his private key, and makes public N and b.

[1] Apparently they came up with this scheme as graduate students, while thinking about how to play poker securely over the phone. In 2013 they received the Turing award for their work in cryptography.

Alice has her message $m = 0$ or 1. She chooses a random integer r such that $1 < r < N$. She then computes the ciphertext

$$c = \begin{cases} r^2 \pmod N & \text{if } m = 0, \\ br^2 \pmod N & \text{if } m = 1. \end{cases}$$

She sends c to Bob.

Bob uses his private key p to compute the Legendre symbol

$$\left(\frac{c}{p}\right) = \begin{cases} \left(\frac{r^2}{p}\right) = 1 & \text{if } m = 0, \\ \left(\frac{br^2}{p}\right) = \left(\frac{b}{p}\right)\left(\frac{r^2}{p}\right) = \left(\frac{b}{p}\right) = -1 & \text{if } m = 1. \end{cases}$$

So he can deduce what the original m was.

On the other hand, Eve, who only has access to N, has to content herself with the Jacobi symbol $\left(\frac{c}{N}\right)$. But, as you will check in this week's tutorial, this turns out to be 1 independently of what m is, so it reveals no information about m.

In terms of practicality, this is a whole lot of trouble just to send one bit securely (so it better be worth it). But the idea of mixing in some randomness into a deterministic cryptosystem is a powerful one.

2.4 Sums of squares

A part of the Quadratic Residue Theorem is: for any odd prime p we have

$$\left(\frac{-1}{p}\right) = (-1)^{\frac{p-1}{2}},$$

which can be restated as: there exists x such that $x^2 \equiv -1 \pmod p$ if and only if $p \equiv 1 \pmod 4$.

A further restatement could be: $x^2 + 1^2 = pk$ for some x, k if and only if $p \equiv 1 \pmod 4$.

We can (and should) be more precise in one direction:

Lemma 2.15. *If $p \equiv 1 \pmod 4$ then there exist x with $1 \leqslant x < p/2$ and k with $1 \leqslant k \leqslant p/4$ such that $x^2 + 1^2 = pk$.*

Proof. As above we know that there exists y with $1 \leqslant y < p$ such that $y^2 \equiv -1 \pmod p$. But $p - y$ satisfies the same properties, and one of $y, p - y$ is $< p/2$; so we let x be the one that satisfies this inequality.

Now there is an integer k such that

$$pk = x^2 + 1 < \frac{p^2}{4} + 1 \quad \Rightarrow \quad pk \leqslant \frac{p^2}{4} \quad \Rightarrow \quad k \leqslant \frac{p}{4}.$$

From $pk = x^2 + 1$ we also see that $k \geqslant 1$. □

Let's generalise the setup a bit and ask: which integers n can be written as a sum of two squares?

We file the following inocuous-looking algebraic identity under "good to know":

Lemma 2.16. *For all (integers) x, y, v, w we have*

$$(x^2 + y^2)(v^2 + w^2) = (xv + yw)^2 + (xw - yv)^2.$$

Proof. Straightforward algebraic manipulation. □

Primes as sums of two squares

We start with the prime case:

Proposition 2.17. *If a prime p is a sum of two squares, then $p = 2$ or $p \equiv 1 \pmod 4$.*

Proof. Suppose p is odd and $p = x^2 + y^2$ for some integers x, y.

First note that $p \nmid xy$: if $p \mid xy$ then $p \mid x$ or $p \mid y$, but from $p = x^2 + y^2$ we see that $p \mid x$ and $p \mid y$, so $p^2 \mid x^2 + y^2 = p$, contradiction.

Now $x^2 + y^2 \equiv 0 \pmod p$, so $x^2 \equiv -y^2 \pmod p$. As $p \nmid y$, y has an inverse modulo p, so we can rewrite the last congruence as $(x/y)^2 \equiv -1 \pmod p$. Therefore -1 is a quadratic residue modulo p, from which we conclude (as above) that $p \equiv 1 \pmod 4$. □

The key to establishing the converse of this result is one of Fermat's major contributions to number theory:

Lemma 2.18 (Infinite Descent)**.** *Let p be prime and suppose there exists k with $1 \leqslant k < p$ such that pk is a sum of two squares: $pk = x^2 + y^2$ for some $x, y \in \mathbf{Z}$. Then p itself is a sum of two squares.*

Proof. If $k = 1$, we are done, so suppose $k \geqslant 2$.

Take $v \equiv x \pmod k$ and $w \equiv y \pmod k$, with $|v|, |w| \leqslant k/2$. (Simply take division by k with remainder, then adjust the remainder to be in the correct range.)

Since $v^2 + w^2 \equiv x^2 + y^2 \equiv 0 \pmod k$, there exists s such that $v^2 + w^2 = sk$. But $|v|, |w| \leqslant k/2$, so $sk = v^2 + w^2 \leqslant k^2/2$, therefore $s \leqslant k/2 < k$.

We should rule out the possibility that $s = 0$: it can only happen if $v = w = 0$, which means that $k \mid x$ and $k \mid y$, so $k^2 \mid (x^2 + y^2) = pk$, so $k \mid p$. Since $k < p$, we conclude that $k = 1$, contradiction.

So $1 \leqslant s < k < p$ and we have

$$(x^2 + y^2)(v^2 + w^2) = psk^2,$$

so by Lemma 2.16

$$(xv + yw)^2 + (xw - yv)^2 = psk^2,$$

hence

$$\left(\frac{xv + yw}{k}\right)^2 + \left(\frac{xw - yv}{k}\right)^2 = ps.$$

Note that $xv + yw \equiv x^2 + y^2 \equiv 0 \pmod k$ and $xw - yv \equiv xy - yx \equiv 0 \pmod k$, so the two quantities in round brackets are indeed integers, and we have successfully written ps as a sum of two squares, with $s < k$.

If $s = 1$, we are done. Otherwise we can repeat this process and make s even smaller. Since this cannot continue forever (we started with a finite number k), we eventually get to p expressed as a sum of squares. □

(If you stare at the proof for long enough you note that it still works if we replace p by an arbitrary $N \in \mathbf{Z}_{\geqslant 1}$, but then we have to require that k satisfies $1 \leqslant k < p(N)$, where $p(N)$ denotes the smallest prime divisor of N.)

Proposition 2.19. *Let p be a prime. If $p = 2$ or $p \equiv 1 \pmod 4$, then p is a sum of two squares.*

Proof. Clear that $2 = 1^2 + 1^2$, so let's assume $p \equiv 1 \pmod 4$.

As we have seen in Lemma 2.15, this implies that there exists $1 \leqslant k \leqslant p/4$ such that pk is a sum of squares: $pk = x^2 + y^2$ for some $x, y \in \mathbf{Z}$.

Then, by Infinite Descent, p itself is a sum of two squares. □

Let's see how Infinite Descent plays out in a concrete case:

Example 2.20. $11^2 + 17^2 = 410 = 10 \cdot 41$. We use this to get an expression for 41 as a sum of squares.

We have $p = 41$, $k = 10$, $x = 11$, $y = 17$. Let $v = 1$, $w = -3$, then $v^2 + w^2 = 1 + 9 = 10$, so $s = 1$.

Since $xv + yw = -40$ and $xw - yv = -50$, the resulting expression is $4^2 + 5^2 = 41$, which is certainly the case.

Putting the two propositions together, we have

Theorem 2.21. *A prime p is a sum of two squares if and only if $p = 2$ or $p \equiv 1 \pmod 4$.*

Integers as sums of two squares

We now consider the more general question of characterising the positive integers n that can be written as $n = x^2 + y^2$ for integers x, y.

We say that $n \in \mathbf{Z}_{\geqslant 1}$ is *squarefree* if the only square dividing n is 1. (Equivalently, there is no prime number p such that $p^2 \mid n$. Equivalently, the prime factorisation of n is $n = p_1 \dots p_r$ with p_j distinct primes.)

Given $n \in \mathbf{Z}_{\geqslant 1}$, there is a largest integer s such that $s^2 \mid n$. Write $n = s^2 n_0$, then n_0 is squarefree, and it is called the *squarefree part of n*.

Theorem 2.22. *An integer $n \in \mathbf{Z}_{\geqslant 1}$ is a sum of two squares if and only if its squarefree part is not divisible by any prime $q \equiv 3 \pmod 4$.*

Proof. Write $n = s^2 n_0$, where n_0 is the squarefree part of n.

Suppose n_0 is not divisible by any prime $q \equiv 3 \pmod 4$. Write $n_0 = p_1 \dots p_r$, where each p_j is either 2 or $1 \pmod 4$. By Theorem 2.21, each p_j is a sum of two squares. By repeated application of Lemma 2.16, so is their product n_0. But if $n_0 = x^2 + y^2$ then $n = s^2 n_0 = s^2(x^2 + y^2) = (sx)^2 + (sy)^2$.

Conversely, suppose n_0 is divisible by some prime $q \equiv 3 \pmod 4$ and $s^2 n_0 = n = x^2 + y^2$ for some integers x, y. Let $k \geqslant 0$ be such that q^k divides s exactly, then q^{2k+1} divides n exactly.

We have $x^2 + y^2 \equiv 0 \pmod q$, so $x^2 \equiv -y^2 \pmod q$. I claim that $q \mid y$. Otherwise, y is invertible modulo q and we can write $(x/y)^2 \equiv -1 \pmod q$, contradicting the fact that -1 is not a quadratic residue modulo q when $q \equiv 3 \pmod 4$.

So $q \mid y$, but $q \mid s^2 n_0$, so $q \mid x^2$, hence $q \mid x$. Therefore $q^2 \mid x^2 + y^2 = s^2 n_0$. Since n_0 is squarefree, we must have $q \mid s$, so $k \geqslant 1$. We can divide $s^2 n_0 = x^2 + y^2$ through by q^2 and get another identity of the same type, $(s')^2 n_0 = (x')^2 + (y')^2$. As n_0 has not changed, we still have $q \mid n_0$, but now $(k-1)$ is such that $q^{k-1} \mid s'$. So we can apply the same argument to go to $k-2$, etc., until we end up with $k = 0$.

That is, we get an identity $t^2 n_0 = z^2 + w^2$ where $q \mid n_0$ but $q \nmid t$. By the same argument applied to $z^2 + w^2 \equiv 0 \pmod q$ we get that $q \mid z$ and $q \mid w$ so $q^2 \mid (z^2 + w^2) = t^2 n_0$, contradiction. □

It follows from this that $n \in \mathbf{Z}_{\geqslant 1}$ is a sum of two squares if and only if its squarefree part n_0 is a sum of two squares.

Sums of four squares

Lemma 2.23. *For any prime p there exist integers x, y, z such that $x^2 + y^2 + z^2 \equiv 0 \pmod p$.*

Proof. For $p = 2$ or $p \equiv 1 \pmod 4$, this follows immediately from Theorem 2.21. (Just take $z = 0$.)

Suppose now that $p \equiv 3 \pmod 4$. Let a be the smallest positive quadratic non-residue modulo p, then $a \geqslant 2$. Since -1 is also a quadratic non-residue, the product $-a$ is a quadratic residue. Let x be such that $x^2 \equiv -a \pmod p$.

Consider $a-1$. By the minimality of a, we know that $a-1$ is a quadratic residue, so let y be such that $y^2 \equiv a-1 \pmod{p}$. So far we have $x^2+y^2 \equiv -a+a-1 \equiv -1 \pmod{p}$, so it remains to take $z=1$. □

Be careful! The Lemma is not saying that every prime is a sum of three squares, only that every prime has a multiple that is a sum of three squares. More on this later, maybe.

We need another weird algebraic identity:

Lemma 2.24. *For all (integers) x, y, z, t, r, u, v, w we have*

$$\begin{aligned}&(x^2+y^2+z^2+t^2)(r^2+u^2+v^2+w^2)\\&\quad=(xr+yu+zv+tw)^2+(xu-yr-zw+tv)^2+(xv+yw-zr-tu)^2+(xw-yv+zu-tr)^2.\end{aligned}$$

Proposition 2.25. *Every prime is a sum of four squares.*

Proof. This uses Fermat's Infinite Descent method.

The starting point is Lemma 2.23, which we interpret as saying that there exist integers x, y, z, t and k such that $pk = x^2+y^2+z^2+t^2$. Moreover, from the proof of Lemma 2.23 we know that we can pick x, y, z, t such that $|x| < p/2$, $|y| < p/2$, $z = 1$, and $t = 0$, giving k such that $pk < \frac{p^2}{4} + \frac{p^2}{4} + 1$, in other words $1 \leqslant k \leqslant p/2$.

If $k = 1$, we're done. So suppose $k > 1$. We manufacture a sum of four squares that gives a strictly smaller multiple of p.

Suppose first that k is odd. Pick r, u, v, w such that $r \equiv x, u \equiv y, v \equiv z, w \equiv t \pmod{k}$ and each has absolute value $< k/2$. (We can do this, with strict inequality, because k is odd.)

Since $r^2+u^2+v^2+w^2 \equiv 0 \pmod{k}$, there exists s such that $r^2+u^2+v^2+w^2 = sk$. As $|r| < k/2$ etc., we get that $sk < 4k^2/4$, so $s < k$.

We cannot have $s = 0$: that would require $r = u = v = w = 0$, so $x \equiv y \equiv z \equiv t \equiv 0 \pmod{k}$, so $k^2 \mid x^2+y^2+z^2+t^2 = pk$, so $k \mid p$. But as $1 \leqslant k \leqslant p/2$, this implies $k = 1$, contradiction.

Hence $1 \leqslant s < k$.

Now the weird identity gives us that

$$\begin{aligned}(pk)(sk) &= (x^2+y^2+z^2+t^2)(r^2+u^2+v^2+w^2)\\&= (xr+yu+zv+tw)^2+(xu-yr-zw+tv)^2\\&\quad+(xv+yw-zr-tu)^2+(xw-yv+zu-tr)^2,\\ps &= \left(\frac{xr+yu+zv+tw}{k}\right)^2+\left(\frac{xu-yr-zw+tv}{k}\right)^2\\&\quad+\left(\frac{xv+yw-zr-tu}{k}\right)^2+\left(\frac{xw-yv+zu-tr}{k}\right)^2.\end{aligned}$$

You can check that each fraction is in fact an integer, so we have expressed ps as a sum of four squares, with $1 \leqslant s < k$.

It remains to deal with the case where k is even, which you will do in a tutorial question. □

Putting the primes together via Lemma 2.24, we obtain

Theorem 2.26 (Lagrange)**.** *Every positive integer is a sum of four squares.*

But wait, there's (always) more

You may have wondered, especially given Lemma 2.23, what happens to sums of three squares. This turns out to have a satisfyingly simple answer: a prime p is a sum of three squares if and only if $p \not\equiv 7 \pmod 8$. More generally, an integer n is a sum of three squares if and only if n is not of the form $4^r(8k+7)$. The proof of these statements is significantly more involved than the cases of two or three squares.

In a different direction, we could ask about primes that can be written as the sum of two cubes. However, seeing that

$$p = x^3 + y^3 = (x+y)(x^2 - xy + y^2),$$

asking for this product to give a prime number puts in a lot of constraints and makes the question somewhat boring.

It becomes interesting again if we allow x and y to be rational numbers, not just integers. This is known as Sylvester's problem, and it is still open, although partial results are known. There's a link to elliptic curves that we may explore when the time is right.