Fixed points for discrete logarithms

Mariana Levin, U. C. Berkeley

Carl Pomerance, Dartmouth College

Suppose that G is a group and $g \in G$ has finite order m. Then for each $t \in \langle g \rangle$ the integers n with $g^n = t$ form a residue class mod m. Denote it by

$$\log_g t.$$

The discrete logarithm problem is the computational task of finding a representative of this residue class; that is, finding an integer n with $g^n = t$.

Finding a discrete logarithm can be *very* easy. For example, say $G = \mathbb{Z}/m\mathbb{Z}$ and $g = 1$. More specifically, say $m = 100$ and $t = 17$. Then $\log_g t = 17$ (or more precisely 17 mod 100).

Lets make it harder: take g as some other generator of $\mathbb{Z}/m\mathbb{Z}$. But then computing $\log_g t$ is really solving the congruence

$$ng \equiv t \bmod m$$

for n, which we've known how to do easily essentially since Euclid.

The cyclic group of order m:

What does this title mean, especially the key word "The"?

Take $G_1 = \mathbb{Z}/100\mathbb{Z}$ and $G_2 = (\mathbb{Z}/101\mathbb{Z})^\times$. Both are cyclic groups of order 100. Both are generated by 3. And 17 is in both groups.

So, there are two versions of computing $\log_3 17$, one in G_1 and one in G_2.

In G_1, we are solving $3n \equiv 17 \bmod 100$. The inverse of 3 is 67, so $n \equiv 17 \cdot 67 \equiv 39 \bmod 100$.

In G_2, we are solving $3^n \equiv 17 \bmod 101$. And this seems much harder.

The moral: when someone talks about *the* cyclic group of a given order, they are not concerned with computational issues.

Well, how can we solve $3^n \equiv 17 \bmod 101$?

Clearly, one way is trial and error, where we compute each power of 3 mod 101 till we find our target 17. The complexity of doing this in a cyclic group of order m is $O(m)$ (and this upper bound also stands as a lower bound for trial and error for most target elements t).

The Diffie–Hellman key-exchange protocol:

Say we have a cyclic group generated by g, which everyone knows. Alice has a secret integer a and "publishes" g^a. Similarly, Bob has a secret integer b and publishes g^b.

Alice and Bob want to set up a secure session with a secret key that only they know, yet they want to set this up over a public line. Here's how they do it: Alice takes Bob's group element g^b and raises it to her secret exponent a, getting $(g^b)^a = g^{ab}$. Bob arrives at the same group element via a different method, namely $(g^a)^b = g^{ab}$.

Eve (an eavesdropper) knows something's afoot and knows g^a and g^b, but apparently cannot easily compute g^{ab} without finding either a or b, that is without solving the dl problem.

So, a group that is well-suited for cryptographic purposes is one where

- it is easy to apply the group operation;
- it is difficult (in practice) to solve the discrete logarithm problem.

Consider the multiplicative group $(\mathbb{Z}/p\mathbb{Z})^\times$, where p is a large prime.

Use the following facts about this group: It is a homomorphic image of semigroup $\mathbb{Z}$ under times. A factorization of an element of $\mathbb{Z}$ coprime to p then maps to a "relation" among group elements.

For example, in $(\mathbb{Z}/101\mathbb{Z})^\times$, we have

$$5^3 \equiv 125 \equiv 24 \equiv 2^3 \cdot 3 \bmod 101, \quad 2^7 \equiv 128 \equiv 27 \equiv 3^3 \bmod 101.$$

Thus,

$$3\log_3 5 \equiv 3\log_3 2 + 1 \bmod 100, \quad 7\log_3 2 \equiv 3 \bmod 100,$$

from which it may be deduced that

$$\log_3 2 \equiv 43 \cdot 3 \equiv 29 \bmod 100, \quad \log_3 5 \equiv 96 \bmod 100.$$

Using $17 \cdot 6 \equiv 1 \bmod 101$, we have

$$\log_3 17 + \log_3 2 + 1 \equiv 0 \bmod 100,$$

so using $\log_3 2 \equiv 29 \bmod 100$

$$\log_3 17 \equiv 70 \bmod 100.$$

This kind of thing can be formalized into the "index calculus" algorithm:

- Choose random numbers r, each time compute g^r mod p, and save any that happen to factor into small primes.

- After enough of these have been saved, we can use linear algebra over the ring $\mathbb{Z}/(p-1)\mathbb{Z}$ to solve for the dl's of the small primes.

- Assuming this is accomplished, again choose random numbers r until one is found where $g^r t$ factors into small primes.

If

$$g^r t \equiv p_1^{a_1} \dots p_k^{a_k} \bmod p,$$

then using the pre-computed numbers $\log_g p_i$, we get

$$\log_g t \equiv -r + a_1 \log_g p_1 + \dots + a_k \log_g p_k \mod (p-1).$$

This kind of idea can be copied for any group which is a homomorphic image of a multiplicative structure where we have factorization into "small" elements. (The set of small elements used is called the "factor base".)

So, for example, the index calculus method can be used in many cases for finding dl's in $\mathbb{F}_q^\times$. Eg, say $q = p^a$, with p prime and a large. We can view $\mathbb{F}_q$ as $\mathbb{F}_p[x]/(f(x))$ where f is irreducible of degree a. And $\mathbb{F}_p[x]$ is a Euclidean domain.

If a is small, we can view $\mathbb{F}_q$ as $\mathcal{O}_K/(p)$ where K is an algebraic number field of degree a over $\mathbb{Q}$ in which p is inert. Even though $\mathcal{O}_K$ may not be a Euclidean domain, and perhaps not even a PID, we do have unique factorization of ideals and we do have a sense of size afforded by the norm. Problems remain, but in many cases the index calculus method is useful.

And there are very important improved versions that employ ideas from the number field sieve for factoring integers.

Thus, cryptographers tend to shy away from the groups $\mathbb{F}_q^\times$.

What generic algorithms might exist other than listing all of the powers of g?

Well, there's "baby steps, giant steps" (known in the CS world as "meet in the middle"):

- Have g of order m and $t \in \langle g \rangle$. Find $k = \lceil \sqrt{m} \rceil$ and g^{-1}.
- Compute the baby steps $tg^0, tg^{-1}, \ldots, tg^{-(k-1)}$ and the giant steps $g^0, g^k, \ldots, g^{(k-1)k}$.
- Sort both lists and find a coincidence between them, say $tg^{-i} = g^{jk}$. Then $t = g^{i+jk}$ and $\log_g t = i + jk$.

Why must there be a coincidence between the two lists?

Well, since $t \in \langle g \rangle$, there is some $n \in [0, m-1]$ with $g^n = t$. Write n in base k, so that since $k^2 > m-1$, we have $n = i + jk$ for some integers $i, j \in [0, k-1]$. And thus, $tg^{-i} = g^{jk}$.

The algorithm presupposes labels for group elements that allows them to be sorted. Sorting can be done in time not much larger than the size of the set to be sorted, and after this, finding the match between the two parts takes only $O(k) = O(\sqrt{m}\,)$ comparisons.

In all, baby steps, giant steps takes $O(\sqrt{m}\log m)$ group operations. It is essentially a universal algorithm, so cryptographers can't avoid it.

A downside of baby steps, giant steps is that it is not so easy to distribute the work to many computers. Another algorithm due to Pollard can be distributed and is what's used in practice to benchmark cryptosystems. It's interesting that Pollard's method is heuristic while baby steps, giant steps is rigorous. Of course, if an answer is found, it is easily checked, so the heuristic part deals with whether the algorithm will terminate within the supposed time bound (which is also about $\sqrt{m}$).

So, can we find a family of convenient groups for which the only dl algorithms take exponential time?

It's hard to prove that it is so, but many people feel that elliptic curve groups over finite fields fit this bill.

But this is for another time, our topic today is not crypto, nor dl algorithms, but fixed points, the equation

$$\log_g x = x.$$

First note that the equation $\log_g x = x$ doesn't make complete sense, since the first "x" is an element of the cyclic group $\langle g \rangle$ and the second x is an integer (or residue class modulo the order of g).

We can make sense by the conflation of integers with residue classes, as we have already been doing. In particular, in the group $(\mathbb{Z}/p\mathbb{Z})^\times$ with generator g, the equation $\log_g x = x$ could be taken to mean that x is an integer in $[1, p-1]$ with $g^x \equiv x \pmod{p}$.

Lets see if such fixed points exist for small primes p:

For $p = 2$, we have $g = 1$, $x = 1$, and yes, $g^x \equiv x \pmod{p}$.

For $p = 3$, we have $g = 2$, and $2^1 \not\equiv 1 \pmod 3$, $2^2 \not\equiv 2 \pmod 3$, so no, there is no fixed point.

For $p = 5$, there are two primitive roots (i.e., cyclic generators for $(\mathbb{Z}/p\mathbb{Z})^\times$), namely 2 and 3. One quickly checks that with the base 3, there are no fixed points, but $2^3 \equiv 3 \pmod 5$.

For $p = 7$, the primitive roots are 3 and 5, and we have

$$3^2 \equiv 2 \pmod 7, \quad 3^4 \equiv 4 \pmod 7, \quad 3^5 \equiv 5 \pmod 7.$$

In Guy, section F9, it is mentioned that D. Brizolis conjectured that for every prime $p > 3$ there is a primitive root g and an integer x in $[1, p-1]$ with $\log_g x = x$.

A necessary and sufficient condition: Suppose $x \in [1, p-1]$ has multiplicative order $(p-1)/d$. There is a primitive root g for p with $\log_g x = x$ if and only if $\gcd(x, p-1) = d$.

Here's why: Let g_0 be a primitive root for p. Every primitive root for p is of the form g_0^i, where i is coprime to $p-1$. Say $g_0^j = x$, so that $(j, p-1) = d$. TFAE:

- There is a primitive root g with $g^x = x$.
- There is an integer i coprime to $p-1$ with $g_0^{ix} = x$.
- There is an integer i coprime to $p-1$ with $ix \equiv j \pmod{p-1}$.
- $\gcd(x, p-1) = \gcd(j, p-1) = d$.

Let us say that a prime p has the "strong Brizolis property" if there is a primitive root g in the range $[1, p-1]$ that is coprime to $p-1$. This is the case $d = 1$ from the previous criterion.

How many such primitive roots do we expect? Well, there are exactly $\varphi(p-1)$ primitive roots in $[1, p-1]$ and exactly $\varphi(p-1)$ integers in this range coprime to $p-1$. If these are "independent events", then we would expect

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

such numbers. Since $\varphi(n) > cn/\log\log n$, the above expression is at least of order $p/(\log\log p)^2$, which is positive for all large p.

How might we try and prove this?

Lets begin with characteristic functions.

Say $f_1(g)$ is 1 if $\gcd(g, p-1) = 1$ and 0 otherwise, and $f_2(g)$ is 1 if g is a primitive root for p and 0 otherwise.

Let $N(p)$ be the number of integers in $[1, p-1]$ that are both primitive roots for p and coprime to $p-1$. Then

$$N(p) = \sum_{g=1}^{p-1} f_1(g) f_2(g).$$

To use this, we need explicit representations for these characteristic functions. Being coprime to $p-1$ is easy, it is essentially a combinatorial inclusion-exclusion over common divisors of g and $p-1$. We have

$$f_1(g) = \sum_{d|\gcd(g,p-1)} \mu(d),$$

where μ is the Möbius function. (We have $\mu(n) = (-1)^{\omega(n)}$ if n is squarefree and 0 otherwise, where $\omega(n)$ is the number of different primes which divide n.)

A combinatorially similar idea works for $f_2(g)$, the characteristic function for primitive roots for p, but here we need to introduce characters. Let g_0 be some primitive root for p and let $\zeta = e^{2\pi i/(p-1)}$, a primitive $(p-1)$st root of 1 in $\mathbb{C}$. There is a natural isomophism χ from $(\mathbb{Z}/p\mathbb{Z})^\times$ to $\langle\zeta\rangle$ where $\chi(g_0^j) = \zeta^j$. Then

$$f_2(g) = \sum_{m|p-1} \frac{\mu(m)}{m} \sum_{j=1}^{m} \chi(g)^{j(p-1)/m}.$$

This can be seen by noting that the inner sum is m if $g^{(p-1)/m} \equiv 1 \pmod{p}$ and 0 otherwise.

So for $N(p)$, the number of integers in $[1, p-1]$ that satisfy the strong Brizolis property for p,

$$N(p) = \sum_{g=1}^{p-1} \sum_{d \mid \gcd(g,p-1)} \mu(d) \sum_{m \mid p-1} \frac{\mu(m)}{m} \sum_{j=1}^{m} \chi(g)^{j(p-1)/m}.$$

Fine, but are we making any progress? It is perhaps natural to write $g = dh$, use $\chi(g) = \chi(d)\chi(h)$ and rearrange a bit. We have

$$N(p) = \sum_{d,m \mid p-1} \frac{\mu(d)\mu(m)}{m} \sum_{j=1}^{m} \chi(d)^{j(p-1)/m} \sum_{h=1}^{(p-1)/d} \chi(h)^{j(p-1)/m}.$$

Note that the terms in this triple sum with $j = m$ are

$$\sum_{d,m \mid p-1} \frac{\mu(d)\mu(m)}{m} \frac{p-1}{d} = \frac{\varphi(p-1)^2}{p-1}.$$

We have proved that

$$\left|N(p)-\frac{\varphi(p-1)^2}{p-1}\right|\le\sum_{d,m|p-1}\frac{|\mu(d)\mu(m)|}{m}\sum_{j=1}^{m-1}\left|\sum_{h=1}^{(p-1)/d}\chi(h)^{j(p-1)/m}\right|.$$

Let

$$S(\chi^{j(p-1)/m})=\max_n\left|\sum_{h=1}^{n}\chi(h)^{j(p-1)/m}\right|,$$

when $1\le j\le m-1$. Thus,

$$\left|N(p)-\frac{\varphi(p-1)^2}{p-1}\right|\le\sum_{d,m|p-1}\frac{|\mu(d)\mu(m)|}{m}\sum_{j=1}^{m-1}S(\chi^{j(p-1)/m}).$$

The Pólya–Vinogradov inequality

In 1918, Pólya and Vinogradov independently showed that for a nonprincipal character ψ modulo q, we have

$$S(\psi) := \max_n \left| \sum_{h=1}^{n} \psi(h) \right| < cq^{1/2} \log q,$$

for a universal positive constant c. Here, ψ is a nontrivial homomorphism from $(\mathbb{Z}/q\mathbb{Z})^\times$ into $\mathbb{C}^\times$ extended to $\mathbb{Z}$ by letting it be 0 when the argument is not coprime to q. Thus,

$$\sum_{d,m|p-1} \frac{|\mu(d)\mu(m)|}{m} \sum_{j=1}^{m-1} S(\chi^{j(p-1)/m}) = O(4^{\omega(p-1)} p^{1/2} \log p),$$

and since $\omega(n) = o(\log n)$, we have the above expression being of magnitude at most $p^{1/2+\epsilon}$.

Thus,

$$N(p) = \frac{\varphi(p-1)^2}{p-1} + O(p^{1/2+\epsilon}).$$

Since as we have seen, the main term is at least of order $p/(\log\log p)^2$, this shows that all sufficiently large primes p have $N(p) > 0$.

But is it true for all primes $p > 3$?

Questions like this pose a computational challenge, since it involves putting explict constants on all of the inequalities involved. And challenges can remain, since the point at which $N(p) > 0$ is proved to be true may be too large to do a case study up to that point.

Some history: W.-P. Zhang in 1995 gave essentially the above argument but did not work out a starting point for when it is true.

C. Cobelli and A. Zaharescu in 1999 gave a somewhat different proof, showing that $N(p) > 0$ for all $p > 10^{2070}$. They said that a reorganization of their estimates would likely support a bound near 10^{50}.

So, can we do better?

An elementary argument shows that

$$4^{\omega(n)} < 1404 n^{1/3}$$

for all natural numbers n. Yes, it is a far cry from n^{ϵ}, but it is explicit, and near to best possible for some numbers n.

Further, one can show that

$$\varphi(n) > \frac{n}{2 \log \log n}$$

for n larger than 1 more than the product of the first 11 primes, about 2×10^{11}.

Earlier this year, I proved that for ψ a nonprincipal, primitive Dirichlet character modulo q, we have

$$S(\psi) = \max_n \left| \sum_{h=1}^{n} \psi(h) \right| \leq q^{1/2} \left(\frac{1}{2\pi} (\log q + 2 \log \log q) + 1 \right).$$

(Note that "primitive" means that the character is not induced by another character to a smaller modulus; for a prime modulus, every nonprincipal character is primitive.)

My proof used some classical Fourier series arguments, a paper of Landau from 1918, and an idea of Bateman as reported in a paper of Hildebrand. (There are other explicit versions of this inequality in the literature, but they are not as sharp.)

So, armed with all of these tools, the bound can be brought down to 10^{38}, but this is still too big to close the gap.

What to do, give up? Not us.

We viewed the formula

$$N(p) = \sum_{d,m|p-1} \frac{\mu(d)\mu(m)}{m} \sum_{j=1}^{m} \chi(d)^{j(p-1)/m} \sum_{h=1}^{(p-1)/d} \chi(h)^{j(p-1)/m}$$

combinatorially as a double inclusion-exclusion, and then used the Bonferroni inequalities. These allow one to truncate an inclusion-exclusion at an odd point to get a lower bound (and at an even point to get an upper bound). So we stop the sum over d when d has an odd number of primes. For each d, we have a full inclusion-exclusion over m, so we stop that at an odd point if the corresponding d has $\mu(d) = 1$ and we stop it at an even point if $\mu(d) = -1$.

Using the Bonferroni inequalities, we greatly reduce the number of terms which involve the Pólya–Vinogradov inequality at a sacrifice of a smaller main term. Below 10^{38} we attacked separately each possible value for $\omega(p-1)$ so that we didn't have to use the one-size-fits-all inequality that we had. Jiggling parameters, we were able to handle each case with $\omega(p-1) \geq 12$.

For $\omega(p-1) = 11$, the argument almost works, and we were able to handle the situation for all but 12 primes p that we could check directly.

For the cases $\omega(p-1) = 10$, 9, and 8, we were able to show that only certain ranges of primes needed to be checked, and then only those satisfying certain congruences, like $p \equiv 1 \pmod{210}$.

We also checked exhaustively up to 6.6×10^9, finding no counterexamples. Our sieve estimates showed that for $\omega(p-1) \leq 7$, any possible counterexample would be below this bound, thus completing the proof.

Finally:

Theorem. (Levin, P) *For every prime $p > 3$ there is a primitive root g for p in $[1, p-1]$ that is coprime to $p-1$. In particular, there is a primitive root g for p and an integer x in $[1, p-1]$ with $\log_g x = x$.*

It is expected that our method of proof can work for other problems involving primitive roots, such as the conjecture of Vegh that for all primes $p > 61$, each residue is the difference of two primitive roots. And some conjectures of Golomb that are similar, but involve more complicated equations than $x - y = z$. There is a fairly wide literature on this group of problems, with papers of Cohen, Le, Mullen, and Sun, but a computational gap remains.