Random primes in arithmetic progressions

Pascal Giorgi
LIRMM, Univ. Montpellier, CNRS
www.lirmm.fr/~giorgi

Bruno Grenet
LIRMM, Univ. Montpellier, CNRS
www.lirmm.fr/~grenet/

Armelle Perret du Cray
LIRMM, Univ. Montpellier, CNRS
www.lirmm.fr/armelle-perret-du-cray

Daniel S. Roche
United States Naval Academy
www.usna.edu/cs/roche

February 11, 2022

Abstract

We describe a straightforward method to generate a random prime q such that the multiplicative group $\mathbb{F}_q^*$ also has a random large prime-order subgroup. The described algorithm also yields this order p as well as a p'th primitive root of unity ω. The methods here are efficient asymptotically, but due to large constants may not be very useful in practical settings.

1 Introduction

In various contexts, for example in sparse polynomial evaluation and interpolation algorithms, it is necessary to have a finite field $\mathbb{F}_q$ that admits an order-p multiplicative subgroup with generator ω. There are typically some non-divisibility properties both on the field size q and the subgroup order p.

In this note, we briefly sketch efficient algorithms to probabilistically generate such q, p, ω tuples. The results are neither surprising to practitioners in this area, nor are they particularly original. However, we have found them useful, and so decided to publish in this short note with complete proofs.

2 Statement of results

We need to find two prime numbers p, q such that $p \mid (q-1)$, that is q is in the arithmetic progression $\{aq+1 : a \geqslant 1\}$, and such that $q = \mathsf{poly}(p)$. Effective versions of Dirichlet's theorem Rousselet (1985); Akbary and Hambrook (2015) allow us to produce such pairs. Theorem 2.1 below is a variant of Lemma 2.4.15 in Arnold's Ph.D. thesis (Arnold, 2016), where we replace a constant probability of success by an arbitrary high probability of success.

Theorem 2.1. *There exists an explicit Monte Carlo algorithm which, given a bound* $\lambda \geqslant \max(2^{74}, \frac{2^{63}}{\epsilon^2})$*, produces a triple* (p, q, ω) *that has the following properties with probability at least* $1 - \epsilon$*, and return* FAIL *otherwise:*

- p *is uniformly distributed amongst the primes of* $(\lambda, 2\lambda)$*;*
- $q \leqslant \lambda^6$ *is a prime such that* $p \mid (q-1)$*;*
- ω *is a* p*-primitive root of unity in* $\mathbb{F}_q$*;*

Its worst-case bit complexity is $\mathsf{polylog}(\lambda)$*.*

An additional requirement in some situations is that the prime q does not divide an (unknown!) large integer. This is achieved by taking λ sufficiently large.

Theorem 2.2. *Let* K *be an unknown integer, and let* (p, q, ω) *a triple produced by the algorithm of Theorem 2.1 on some input* λ*. If* $\lambda \geqslant \max(\frac{2^{61}}{\mu^2}, \sqrt[5]{\frac{48}{\mu} \ln K})$*, the probability that* q *divides* K *is at most* μ*.*

We note that, unfortunately, the algorithm of Theorem 2.1 is not very practical due to the large (but constant!) lower bound on the bit-length of the primes produced. In practice a far simpler and more efficient approach works much better: Simply choose a random b-bit prime p, then try each $k = 1, 2, 3, \ldots$ until $2pk + 1$ is prime. A conjecture of Heath-Brown (1978) states that $k \ll \log^2 p$ for any prime p. (Later work by Granville and Pomerance (1990) conjectures further that this bound is asymptotically tight in the worst case.) Under this conjecture, the simple enumerative approach above always finds the least prime q congruent to 1 modulo p in $\mathsf{poly}(b)$ time, and this is the most effective technique in practice.

3 Proofs

To construct a field $\mathbb{F}_q$ with a p-PRU ω, we first need to generate random prime numbers. The well-known technique for this is to sample random integers and test them for primality. In order to get Las Vegas algorithm, we rely on the celebrated AKS algorithm.

Fact 3.1 (Agrawal, Kayal, and Saxena (2004))**.** *There is a deterministic algorithm that, given any integer* n*, determines whether* n *is prime or composite and has bit complexity* $\mathsf{polylog}(n)$*.*

While the original bit complexity was $\widetilde{O}\big(\log^{10.5} n\big)$, this has been subsequently improved to $\widetilde{O}\big(\log^6 n\big)$ in a revised version by Lenstra and Pomerance (2011). In practice, a better option is to use the Monte Carlo Miller-Rabin primality test which has a worst-case bit complexity of $\widetilde{O}\big(\log^2 n\big)$ but a low probability of incorrectly reporting that a composite number is prime (Rabin, 1980).

No fast deterministic algorithm is known to *construct* a prime number with a given bit length b. However, sampling random b-bit integers and testing their primality using AKS algorithm results in a *Las Vegas* randomized algorithm. The expected running time relies on the fact that there are at least $\Omega(2^b/b)$ primes with b bits. We recall some more precise bounds.

Fact 3.2 (Rosser and Schoenfeld (1962))**.** *For $\lambda \geqslant 21$, there exist at least $\frac{3}{5}\lambda \ln \lambda$ prime numbers between λ and 2λ.*

Once we have a prime number p, we want to find a prime number q in the arithmetic progression $p+1$, $2p+1$, $3p+1$, ... Dirichlet's theorem says that, *asymptotically*, the distribution of primes in this arithmetic progression is the same as the distribution of primes in $\mathbb{Z}$. This indicates that a good strategy to generate q is simply to pick a random (even) positive integer k and test whether $pk+1$ is prime, repeating until a prime of that form is found.

The question is, how large should k be in the strategy above in order to guarantee a reasonable chance of success? Linnik's theorem Linnik (1944) states that there exists a constant $1 < L \leqslant 5$ such that for all sufficiently large primes p, choosing $k \sim p^{L-1}$ is enough. On the other hand, Rousselet (1985) showed that choosing $k \sim p^2$ will work for *most* primes p that are large enough. Using more recent results by Akbary and Hambrook (2015), it is possible to obtain completely explicit bounds.

Fact 3.3. *Let $0 < \epsilon < 1$ and $\lambda \geqslant \max(2^{74}, \frac{2^{59}}{\epsilon^2})$, and p a random prime from $(\lambda, 2\lambda)$. Then with probability at least $1-\epsilon$, the number of prime numbers $q \leqslant \lambda^6$ of the form $q = ap+1$ is $\geqslant \lambda^5/(24 \ln \lambda)$.*

Proof. Let $\pi(x)$ denote the number of prime numbers $\leqslant x$, $\pi(x; m, a)$ the number of prime numbers $\leqslant x$ that are congruent to a modulo m, and $\ell(x)$ the smaller prime divisor of x. Akbary and Hambrook (2015, Corollary 1.4) prove that for any $\lambda_1 \leqslant \lambda_2 \leqslant \gamma^{1/2}$,

$$\sum_{\substack{m \leqslant \lambda_2 \\ \ell(m) > \lambda_1}} \max_{2 \leqslant y \leqslant \gamma} \max_{a : \gcd(a,m)=1} \left| \pi(y; m, a) - \frac{\pi(y)}{\phi(m)} \right|$$
$$\leqslant 346.21 \left(4\frac{\gamma}{\lambda_1} + 4\gamma^{1/2}\lambda_2 + 18\gamma^{2/3}\lambda_2^{1/2} + 5\gamma^{5/6} \ln(e\frac{\lambda_2}{\lambda_1}) \right) (\ln \gamma)^{9/2}.$$

We apply this inequality with $\lambda_1 = \lambda$, $\lambda_2 = 2\lambda$ and $\gamma = \lambda^6$. We note that the sum is over the prime numbers (since $\ell(m) > \lambda_1 \geqslant m/2$). We then simplify it by choosing $y = \gamma$ and $a = 1$ in the formula, which can only make the sum smaller. Then

$$\sum_{\substack{\lambda < p < 2\lambda \\ p \text{ prime}}} \left| \pi(\lambda^6; p, 1) - \frac{\pi(\lambda^6)}{p-1} \right| \leqslant 1.38 \cdot 10^7 (\lambda^5 + 2.03\lambda^{4.5} + 0.64\lambda^4)(\ln \lambda)^{9/2}$$

For $\lambda \geqslant 2^{15}$, the sum is bounded by $1.4 \cdot 10^7 \lambda^5 (\ln \lambda)^{9/2}$. Now we wish to count the *bad* primes in $(\lambda, 2\lambda)$ such that $\pi(\lambda^6; p, 1) \leqslant \lambda^5/(24 \ln \lambda)$. Since $\pi(\lambda^6) \geqslant$

$\lambda^6/(6\ln\lambda)$, if p is a bad prime, then $\pi(\lambda^6)/(p-1) \geqslant \pi(\lambda^6;p,1)$ and since $p-1 \leqslant 2\lambda$,

$$\left|\pi(\lambda^6;p,1) - \frac{\pi(\lambda^6)}{p-1}\right| \geqslant \frac{\lambda^6/(6\ln\lambda)}{p-1} - \frac{\lambda^5}{24\ln\lambda} \geqslant \frac{\lambda^5}{24\ln\lambda}.$$

If there are k bad primes, then the sum is at least $k\lambda^5/24\ln\lambda$. Using the previous bound on the sum, we get the bound

$$k \leqslant \frac{1.4\cdot 10^7\lambda^5(\ln\gamma)^{9/2}}{\lambda^5/(24\ln\lambda)} = 3.36\cdot 10^8(\ln\lambda)^{11/2}.$$

Since there are at least $\frac{3}{5}\lambda/\ln\lambda$ prime numbers between λ and 2λ, the probability that a random prime number p chosen in $(\lambda, 2\lambda)$ is bad is at most

$$\frac{3.36\cdot 10^8(\ln\lambda)^{11/2}}{\frac{3}{5}\lambda/\ln\lambda} = 5.6\cdot 10^8\lambda^{-1}(\ln\lambda)^{13/2}.$$

The probability obviously tends to zero when λ tends to infinity, as $O((\ln\lambda)^{13/2}/\lambda)$. Therefore, to get a probability $\leqslant \epsilon$, one should consider $\lambda = 1/\epsilon^{\Omega(1)}$. For instance, for $\lambda \geqslant 2^{74}$ the probability is bounded by $5.6\cdot 10^8\lambda^{-1/2}$. Hence, to get a probability at most ϵ, one can take $\lambda \geqslant \max(2^{74}, \frac{2^{59}}{\epsilon^2})$ since $2^{59} > (5.6\cdot 10^8)^2$. □

From this effective result, we deduce a Monte Carlo algorithm that produces primes p, q such that $p \mid (q-1)$, as well as a p-PRU modulo q.

Proof of Theorem 2.1. This is basically Algorithm "GetPrimeAP-5/6" on page 35 of (Arnold, 2016), slightly adapted, where the primality tests are made using AKS algorithm:

1. sample $\leqslant \frac{5}{6}\ln\frac{4}{\epsilon}\ln\lambda$ random odd integers $p \in (\lambda, 2\lambda)$ until p is prime, return FAIL if none of them is prime
2. sample $\leqslant 24\ln\frac{4}{\epsilon}\ln\lambda$ random integers $a \in [1, \lambda^5]$ until $q = ap+1$ is prime, return FAIL if none of them is prime
3. sample $\leqslant \log_p\frac{4}{\epsilon}$ random elements $\zeta \in \mathbb{F}_q^\times$ until $\omega = \zeta^{(q-1)/p} \neq 1$, return FAIL if $\omega = 1$ for each ζ
4. **return** (p, q, ω)

Since AKS has complexity $\mathsf{polylog}\lambda$ and $\log\frac{1}{\epsilon} = O(\log\lambda)$, the complexity of the whole algorithm is $\mathsf{polylog}(\lambda)$.

There are at least $\frac{3}{5}\lambda/\ln\lambda$ primes in $(\lambda, 2\lambda)$, and $\lambda/2$ odd integers. Therefore, the probability that a random odd integer is prime is at least $6/(5\ln\lambda)$. The probability that no prime is produced after k tries is at most $(1-6/(5\ln\lambda))^k \leqslant e^{-6k/(5\ln\lambda)}$. If $k = \frac{5}{6}\ln\frac{4}{\epsilon}\ln\lambda$, the probability is at most $\frac{\epsilon}{4}$. Hence Step 1 succeeds with probability at least $1-\frac{\epsilon}{4}$.

Since $\lambda \geqslant \max(2^{74}, \frac{2^{59}}{(\epsilon/4)^2})$, if the algorithm succeeds in producing p, there are at least $\lambda^5/(24\ln\lambda)$ prime numbers $q \leqslant \lambda^6$ of the form $ap+1$ with probability at least $\frac{\epsilon}{4}$.

If p satisfies this condition, there are at least $\lambda^5/(24\ln\lambda)$ values of a such that $ap+1$, amongst the λ^5 possible values. With the same proof as above, the probability that such an a be found in $\leqslant 24\ln\frac{4}{\epsilon}\ln\lambda$ tries is at least $1-\epsilon 4$.

Finally, if q has been found, Step 3 finds a suitable ω with probability at least $1-\frac{\epsilon}{4}$ since there are at most $\frac{q-1}{p}$ values of ζ such that $\zeta^{(q-1)/p}=1$.

Therefore, the algorithm returns a triple (p,q,ω) satisfying the three properties with probability at least $1-\epsilon$. □

Proof of Theorem 2.2. Since $\lambda \geqslant \frac{2^{59}}{(\mu/2)^2}$, the prime p, if produced, satisfies that there are at least $\lambda^5/(24\ln\lambda)$ primes $q\leqslant\lambda^6$ of the form $ap+1$ with probability at least $1-\frac{\mu}{2}$. The number of those primes than can divide K is at most $\log_\lambda K$ (all of them are $>\lambda$). Therefore, the probability that one of them chosen at random divides K is at most $24\log_p K\ln\lambda/\lambda^5\leqslant\frac{\mu}{2}$. □

References

Manindra Agrawal, Neeraj Kayal, and Nitin Saxena. 2004. PRIMES is in P. *Annals of Mathematics* 160, 2 (2004), 781–793. https://doi.org/10.4007/annals.2004.160.781 Referenced on page 2.

Amir Akbary and Kyle Hambrook. 2015. A variant of the Bombieri-Vinogradov theorem with explicit constants and applications. *Math. Comp.* 84, 294 (2015), 1901–1932. https://doi.org/10.1090/S0025-5718-2014-02919-0 Referenced on pages 1 and 3.

Andrew Arnold. 2016. *Sparse Polynomial Interpolation and Testing*. Ph.D. Dissertation. University of Waterloo. http://hdl.handle.net/10012/10307 Referenced on pages 1 and 4.

Andrew Granville and Carl Pomerance. 1990. On the Least Prime in Certain Arithmetic Progressions. *Journal of the London Mathematical Society* s2-41, 2 (1990), 193–200. https://doi.org/10.1112/jlms/s2-41.2.193 Referenced on page 2.

D. R. Heath-Brown. 1978. Almost-primes in arithmetic progressions and short intervals. *Mathematical Proceedings of the Cambridge Philosophical Society* 83, 03 (1978), 357–375. https://doi.org/10.1017/S0305004100054657 Referenced on page 2.

H. W. Lenstra, Jr. and Carl Pomerance. 2011. Primality testing with Gaussian periods. In *FST TCS 2002: Foundations of Software Technology and Theoretical Computer Science*, Manindra Agrawal and Anil Seth (Eds.). Springer Berlin Heidelberg. https://doi.org/10.1007/3-540-36206-1_1 Referenced on page 2.

U. V. Linnik. 1944. On the Least Prime in an Arithmetic Progression. I. The Basic Theorem. *Rec. Math. [Mat. Sbornik] N.S.* 15, 2 (1944), 139–178. http://mi.mathnet.ru/msb6196 Referenced on page 3.

Michael O. Rabin. 1980. Probabilistic algorithm for testing primality. *Journal of Number Theory* 12, 1 (1980), 128–138. https://doi.org/10.1016/0022-314X(80)90084-0 Referenced on page 2.

J. Barkley Rosser and Lowell Schoenfeld. 1962. Approximate formulas for some Functions of Prime Numbers. *Illinois Journal of Mathematics* 6 (1962), 64–94. https://doi.org/10.1215/ijm/1255631807 Referenced on page 3.

Bruno Rousselet. 1985. Estimations du type Brun-Titchmarsh. *Groupe d'étude en théorie analytique des nombres* 1, 37 (1985), 1. Referenced on pages 1 and 3.