

Generating Large Primes

Lecture 12: Random sampling and primality testing

Lecture overview

Random sampling produces large primes efficiently. The strong Miller–Rabin test keeps the chance of returning a composite small.

Lecture slides and worked examples.

Primary source: Lecture 12 [Maj26]

Abstract and lecture roadmap

We find large primes by sampling integers and testing each candidate. The probability of success controls the waiting time.

1. How often is a random n -bit integer prime?
2. Why does the Fermat test miss Carmichael numbers?
3. How does Miller–Rabin control the generator's error?

The recurring proof plan

Give an algorithm, identify the event on which it succeeds, and count the relevant candidates or witnesses.

Based on the supplied Lecture 12 [Maj26]; background: [HPS14, §3.4], [BS23, §16.4.2.2].

Main results

Sampling n -bit integers

Write p_n for the prime probability. The prime number theorem gives

$$p_n \sim \frac{1}{n \log 2}, \quad \mathbb{E}T = p_n^{-1} \sim n \log 2.$$

With an exact test, the chance of no prime in t draws is $(1 - p_n)^t$.

Testing an odd candidate

One Miller–Rabin round always accepts a prime. For every odd composite, a uniformly random base passes with probability at most $1/4$. With r independent bases, this falls to 4^{-r} .

Inputs: Shoup [Sho08, Thms. 5.14 and 10.3]; Hoffstein–Pipher–Silverman [HPS14, §3.4].

An eight-number experiment

The 4-bit integers are

$$\{8, 9, 10, 11, 12, 13, 14, 15\}.$$

The primes in this set are 11 and 13.

If we sample uniformly and repeat until we see a prime, then

$$\Pr(\text{one draw succeeds}) = \frac{2}{8} = \frac{1}{4}, \quad \mathbb{E}(\text{number of draws}) = 4.$$

The same construction works for large n . We need a prime-density estimate and a fast primality test.

Composing the two algorithms

The output's error probability

If the candidate is sampled afresh each time and each composite passes the r -round test with probability at most $\varepsilon = 4^{-r}$, then

$$\Pr(\text{eventual output is composite}) \leq \frac{(1 - p_n)\varepsilon}{p_n + (1 - p_n)\varepsilon} \leq \frac{\varepsilon}{p_n}.$$

The extra factor $1/p_n$ accounts for the many candidates tested before the generator stops. We will prove this bound after building the primality test.

Compare Shoup [Sho08, §9.4.1, pp. 292–294].

The rejection sampler

Let

$$I_n = \{2^{n-1}, 2^{n-1} + 1, \dots, 2^n - 1\}, \quad n \geq 3.$$

GeneratePrime(n), assuming an exact test

1. Draw X uniformly from I_n .
2. If $\text{IsPrime}(X)$, return X .
3. Otherwise, draw a new, independent X .

There are exactly 2^{n-1} candidates. The first draw is not required to be prime: repetition supplies the construction.

Boneh–Shoup [BS23, §16.4.2.2, pp. 699–700].

Why is the output uniform among primes?

Let $M = 2^{n-1}$, let K be the number of primes in I_n , and put $p_n = K/M$.

For a fixed prime $q \in I_n$, the chance that the first success occurs on draw j and equals q is

$$(1 - p_n)^{j-1} \frac{1}{M}.$$

Summing the geometric series,

$$\Pr(\text{output} = q) = \sum_{j \geq 1} (1 - p_n)^{j-1} \frac{1}{M} = \frac{1}{Mp_n} = \frac{1}{K}.$$

Every n -bit prime has exactly the same output probability.

How many n -bit primes are there?

Write $\pi(x) = |\{q \leq x : q \text{ is prime}\}|$. For $n \geq 3$, the number of primes in I_n is

$$K_n = \pi(2^n) - \pi(2^{n-1}).$$

The prime number theorem states $\pi(x) \sim x / \log x$. In the dyadic interval this yields

$$K_n \sim \frac{2^{n-1}}{n \log 2}, \quad p_n = \frac{K_n}{2^{n-1}} \sim \frac{1}{n \log 2}.$$

Here $\log$ is the natural logarithm. The shorthand $1/n$ gives the right order, but misses $1/\log 2$.

Shoup [Sho08, Thm. 5.14, p. 116]; Hoffstein–Pipher–Silverman [HPS14, Thm. 3.21, p. 133].

Waiting time is geometric

Let T be the first draw that is prime. Independent, uniform candidates give

$$\Pr(T > t) = (1 - p_n)^t \leq \exp(-p_n t), \quad \mathbb{E}T = \frac{1}{p_n} \sim n \log 2.$$

The inequality uses $1 - u \leq e^{-u}$ for $0 \leq u \leq 1$.

In particular, for fixed $\alpha > 0$,

$$\Pr(T > \lfloor \alpha n \rfloor) \longrightarrow \exp\left(-\frac{\alpha}{\log 2}\right).$$

The exponent scales as t/n . An e^{-t} tail would be far too strong for a random n -bit candidate.

A 1000-bit estimate

A 1000-bit candidate lies in

$$[2^{999}, 2^{1000} - 1].$$

The large- n estimate gives

$$p_{1000} \approx \frac{1}{1000 \log 2} \approx 0.00144, \quad \mathbb{E} T \approx 1000 \log 2 \approx 693.$$

Screening out even candidates can reduce the number of full primality tests. The displayed estimate concerns uniform sampling from *all* 1000-bit integers [BS23, §16.4.2.2].

What a primality test must guarantee

An *exact* test reports the correct answer on every input. Agrawal, Kayal, and Saxena gave a deterministic algorithm polynomial in the input bit length [AKS04].

A randomized, one-sided test has the following guarantees:

	accepts as prime	rejects as composite
N prime	1	0
N composite	$\leq \varepsilon$	$\geq 1 - \varepsilon$

Rejection proves compositeness. Acceptance is evidence whose strength depends on ε .

Fermat's little theorem

Elementary fact

If p is prime and $1 \leq a < p$, then

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

Proof roadmap. Multiplication by a permutes the nonzero residues modulo p . Multiply those residues before and after the permutation:

$$(p-1)! \equiv a^{p-1}(p-1)! \pmod{p}.$$

Every factor in $(p-1)!$ is invertible modulo p , so cancel the product. □

The atomic Fermat test

For odd $N \geq 3$, sample a uniformly from $\{1, \dots, N-1\}$. Compute $a^{N-1} \bmod N$.

Decision

Pass if $a^{N-1} \equiv 1 \pmod{N}$; otherwise, report that N is composite.

Every prime passes. For example, $2^{14} \equiv 4 \pmod{15}$, so base 2 rejects 15.

A composite may pass. Such a passing base is called a *Fermat liar*; a rejecting base is a *witness*. We first count liars for non-Carmichael composites.

Non-Carmichael composites: half the bases reject

Let N be composite and

$$U_N = (\mathbb{Z}/N\mathbb{Z})^\times, \quad K_N = \{a \in U_N : a^{N-1} = 1\}.$$

The map $a \mapsto a^{N-1}$ is a homomorphism on the abelian group U_N , so K_N is a subgroup.

If N is *not* Carmichael, some unit fails Fermat's congruence. Hence $K_N \neq U_N$, and Lagrange's theorem gives

$$|K_N| \leq \frac{|U_N|}{2} = \frac{\varphi(N)}{2} \leq \frac{N-1}{2}.$$

Nonunits also fail. Thus at least half of the $N-1$ bases reject.

Compare Shoup [Sho08, §10.1].

Why 561 fools the Fermat test

A Carmichael number is a composite N with $a^{N-1} \equiv 1 \pmod{N}$ for every unit a .

Take $561 = 3 \cdot 11 \cdot 17$. The numbers 2, 10, 16 all divide 560. By Fermat's theorem modulo each of 3, 11, 17, every unit a satisfies $a^{560} \equiv 1$ modulo each prime, hence modulo 561.

Exactly

$$\varphi(561) = (3 - 1)(11 - 1)(17 - 1) = 320$$

of the 560 sampled bases pass. The pass fraction is $320/560 = 4/7$, including base 2.

Hoffstein–Pipher–Silverman [HPS14, §3.4, p. 130].

The missing check: square roots of 1

If p is prime, then

$$x^2 \equiv 1 \pmod{p} \implies (x-1)(x+1) \equiv 0 \pmod{p}.$$

Because $\mathbb{Z}/p\mathbb{Z}$ has no zero divisors, $x \equiv \pm 1 \pmod{p}$.

For a composite modulus the implication can fail. In fact,

$$67^2 \equiv 1 \pmod{561}, \quad 67 \not\equiv \pm 1 \pmod{561}.$$

Fermat only inspects the final power a^{N-1} . Miller–Rabin inspects the squaring steps that lead to it.

The strong Miller–Rabin round

For an odd $N \geq 3$, write

$$N - 1 = 2^s d, \quad d \text{ odd}, \quad s \geq 1.$$

Sample a uniformly from $1, \dots, N - 1$. If $\gcd(a, N) > 1$, reject. Otherwise set $x = a^d \bmod N$.

When a base passes

Pass if $x = 1$, or if one of

$$a^d, a^{2d}, a^{4d}, \dots, a^{2^{s-1}d}$$

equals $-1 \pmod{N}$. Reject in every other case.

Obtain successive powers by squaring, so a round takes polynomial time in $\log N$.

Hoffstein–Pipher–Silverman [HPS14, Table 3.2, p. 131].

Why every prime passes

Let $N = p$ be an odd prime, and put $y = a^d \bmod p$. Fermat's theorem gives $y^{2^s} = 1$.

If $y = 1$, the round passes. Otherwise, follow

$$y, y^2, y^4, \dots, y^{2^s} = 1$$

and stop at the *first* 1. Its predecessor squares to 1 but is not 1. Over $\mathbb{Z}/p\mathbb{Z}$, that predecessor must be -1 .

Its exponent is $2^j d$ for some $0 \leq j < s$, so the strong round passes. This proves one-sided completeness.

The same 561, now with base 2

Since $560 = 2^4 \cdot 35$, the Miller–Rabin round checks

exponent	35	70	140	280
$2^{\text{exponent}} \bmod 561$	263	166	67	1

None of the checked residues is 1 at the beginning or $-1 = 560$ at any stage. The round *rejects* 561.

The last two entries display the failure directly: $67^2 \equiv 1 \pmod{561}$, but $67 \not\equiv \pm 1 \pmod{561}$. Nevertheless $2^{560} \equiv 1 \pmod{561}$, so Fermat passes.

Hoffstein–Pipher–Silverman [HPS14, Ex. 3.19, p. 132].

The strong-liar theorem

Rabin's bound

For every odd composite N , at most

$$\frac{N-1}{4}$$

of the $N-1$ bases in $\{1, \dots, N-1\}$ pass a strong Miller–Rabin round.

Proof roadmap. Factor N into prime powers, use the Chinese remainder theorem to describe the bases that pass at each squaring level, and compare the resulting kernel sizes. The prime-power case is handled separately.

We use the counting theorem as a technical input here. Its precise proof location is recorded in the final section.

Independent rounds amplify the guarantee

Choose r independent, uniform bases. Accept N only if every strong round passes.

If N is prime, all rounds pass. If N is an odd composite, the strong-liar theorem gives

$$\Pr(\text{MR}_r(N) \text{ accepts}) \leq \left(\frac{1}{4}\right)^r = 4^{-r} = 2^{-2r}.$$

Even integers $N > 2$ are rejected before the rounds. Each round uses fresh randomness; repeated use of one fixed base would not yield this product bound.

The generator with a randomized test

Sample independent $X \in I_n$ until $\text{MR}_r(X)$ accepts. Handle $X = 2$ directly, and reject even $X > 2$ directly.

Let

$$p = p_n = \Pr(X \text{ prime}),$$

$$q = \Pr(X \text{ composite and } \text{MR}_r(X) \text{ accepts}).$$

The primality guarantee gives $q \leq (1 - p)4^{-r}$.

A trial returns a prime with probability p , returns a composite with probability q , and repeats otherwise. Therefore its expected number of trials is

$$\frac{1}{p + q} \leq \frac{1}{p} = O(n).$$

Proof of the generator's error bound

Write $h = 1 - p - q$ for the probability that one trial asks us to repeat. A composite is returned on trial j with probability $h^{j-1}q$. Hence

$$\begin{aligned}\Pr(\text{output composite}) &= \sum_{j \geq 1} h^{j-1}q \\ &= \frac{q}{p + q} \\ &\leq \frac{(1 - p)4^{-r}}{p + (1 - p)4^{-r}} \leq \frac{4^{-r}}{p}.\end{aligned}$$

The middle inequality uses that $x/(p + x)$ increases in $x \geq 0$. This is the bound stated in the introduction.

Choosing the number of rounds

Suppose the prime-density estimate gives $p_n \geq c/n$ for the bit lengths under consideration, with a fixed $c > 0$.

To make the generator's composite-output probability at most $2^{-\lambda}$, it suffices to choose

$$r \geq \frac{\lambda + \log_2(n/c)}{2}.$$

Then $4^{-r}/p_n \leq 2^{-\lambda}$.

The number of tests is $O(n)$ in expectation. Each test has r modular-exponentiation rounds, polynomial in n and r .

What the two guarantees say

Guarantee	Interpretation
Fixed composite N	$\Pr(\text{MR}_r(N) \text{ accepts}) \leq 4^{-r}$. Probability concerns the test's random bases.
Generator's output	$\Pr(\text{output composite}) \leq 4^{-r}/p_n$. Probability concerns the eventual output.

These are different experiments. A test's error bound cannot be used unchanged as the generator's error bound.

Shoup [Sho08, §9.4.1, pp. 292–294].

Takeaways

1. Random n -bit integers are prime with probability asymptotic to $1/(n \log 2)$.
2. An exact rejection sampler outputs a uniform n -bit prime in $n \log 2 + o(n)$ expected draws.
3. Fermat's test has a Carmichael obstruction; the strong Miller–Rabin round checks the squaring chain.
4. With r independent rounds, every odd composite passes with probability at most 4^{-r} .
5. For the generator, budget the accumulated error through the factor $1/p_n$.

Prime number theorem

As $x \rightarrow \infty$, $\pi(x) \sim x / \log x$. We use the consequence

$$\pi(2^n) - \pi(2^{n-1}) \sim \frac{2^{n-1}}{n \log 2}, \quad p_n \sim \frac{1}{n \log 2}.$$

Exact references. Shoup [Sho08, Thm. 5.14, p. 116]; Hoffstein–Pipher–Silverman [HPS14, Thm. 3.21, p. 133]. For a quantitative dyadic lower bound, see Shoup [Sho08, Thm. 5.8, p. 109].

All geometric-distribution and output-uniformity arguments in the lecture were proved directly.

Technical inputs: primality testing

Strong-liar bound

For odd composite N , a uniform base in $\{1, \dots, N - 1\}$ passes Miller–Rabin with probability at most $1/4$, with $\gcd(a, N) > 1$ counted as rejection.

Exact reference. Shoup [Sho08, §10.2, Thm. 10.3, pp. 309–310, especially (10.1)–(10.4)]. See also Hoffstein–Pipher–Silverman [HPS14, Prop. 3.18, p. 131], which states the bound and omits the proof, and Rabin’s original paper [Rab80].

The deterministic polynomial-time primality theorem mentioned earlier is Agrawal–Kayal–Saxena [AKS04, pp. 781–793].

Technical inputs: elementary algebra

Congruences and groups

CRT. Residues modulo pairwise coprime m_i specify a unique residue modulo $\prod_i m_i$.

Lagrange. If $H \leq G$ and G is finite, then $|H|$ divides $|G|$.

Shoup [Sho08, Thm. 2.6, pp. 23–24; Thm. 6.15, p. 139]. The no-zero-divisors fact for $\mathbb{Z}/p\mathbb{Z}$ is in [Sho08, Ex. 7.19, p. 171].

Fast arithmetic

Euclid's algorithm computes $\gcd(a, N)$, and repeated squaring computes $a^d \bmod N$, in time polynomial in their bit lengths.

Shoup [Sho08, Thm. 4.2, p. 76; §3.4, pp. 65–66].

Further reading

1. Hoffstein–Pipher–Silverman [HPS14, §3.4, pp. 128–136] explains Fermat, Carmichael numbers, Miller–Rabin, and prime density.
2. Boneh–Shoup [BS23, §16.4.2.2, pp. 699–700] discusses sampling primes for RSA and preliminary sieving.
3. Shoup [Sho08, §§9.4, 10.2] develops the generator's error accounting and proves the strong-liar bound.

Suggested exercise: verify the four residues in the 561 example and identify its nontrivial square root of 1.

References I

- [AKS04] Manindra Agrawal, Neeraj Kayal, and Nitin Saxena.
PRIMES is in P.
Annals of Mathematics, 160(2):781–793, 2004.
doi:10.4007/annals.2004.160.781.
- [BS23] **Dan Boneh and Victor Shoup.**
A graduate course in applied cryptography, 2023.
Version 0.6, January 2023.
URL: https://crypto.stanford.edu/~dabo/cryptobook/BonehShoup_0_6.pdf.
- [HPS14] **Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman.**
An Introduction to Mathematical Cryptography.
Undergraduate Texts in Mathematics. Springer, New York, 2 edition, 2014.
doi:10.1007/978-1-4939-1711-2.

References II

- [Maj26] **Hemanta K. Maji.**
Lecture 12: Generating large primes.
CS 355 course slides, 2026.
Supplied lecture slides, 15 pages.
- [Rab80] **Michael O. Rabin.**
Probabilistic algorithm for testing primality.
Journal of Number Theory, 12(1):128–138, 1980.
doi:10.1016/0022-314X(80)90084-0.
- [Sho08] **Victor Shoup.**
A Computational Introduction to Number Theory and Algebra.
Cambridge University Press, 2 edition, 2008.
Author's version 2.5.
URL: https://www.shoup.net/ntb/ntb-v2_5.pdf.