Computational Number Theory and Algebra May 16, 2012

Lecture 9

Lecturers: Markus Bläser, Chandan Saha *Scribe: Chandan Saha*

In the last class, we mentioned that an irreducible polynomial of degree n over a finite field $\mathbb{F}_q$ can be used to generate the extension field $\mathbb{F}_{q^n}$. This gives us a method to construct large finite fields starting from small fields. To give you an example as to where such extension fields are useful, recall that in the Reed-Solomon encoding procedure, we need to use a finite field whose size is at least as large as the codeword length. On the other hand, in the list decoding phase we need to factor a bivariate polynomial. Given that bivariate factoring reduces to univariate factoring and that we only know of a *deterministic* poly-time factoring algorithm for low-characteristic finite fields, it makes sense to start with a small prime field and extend it suitably to a sufficiently large finite field. In today's class, we will see how to generate an irreducible polynomial over a finite field in random polynomial time. The topics of discussion for today's class are:

- Generating irreducible polynomials over finite fields,
- Miller-Rabin primality test.

1 Generating irreducible polynomials over finite fields

We want to generate an irreducible polynomial of degree n over a finite field $\mathbb{F}_q$. Recall from the last class that irreducibility of a given polynomial can be checked in deterministic polynomial time. Now, if we can show that the density of irreducible polynomials is sufficiently large then we can just pick a random polynomial of degree n and test if it is irreducible. This should yield an irreducible polynomial with high probability (provided the density is large). To make this idea formal, we need to estimate the density of irreducible polynomials of degree n over a finite field $\mathbb{F}_q$.

1.1 Density of irreducible polynomials

Lemma 1 *Let $\mathcal{I}(n,q)$ be the number of* monic *irreducible polynomials of degree n over $\mathbb{F}_q$. Then,*

$$\frac{q^n - 2q^{n/2}}{n} \leq \mathcal{I}(n,q) \leq \frac{q^n}{n}.$$

Proof By Lemma 5 of lecture 7, the factors of the polynomial $x^{q^n} - x$ over $\mathbb{F}_q$ are exactly those monic irreducible polynomials (over $\mathbb{F}_q$) whose degree divide n. Let P_d be the product of all monic irreducible factors of degree d. Then, $q^n = \sum_{d|n} \deg(P_d)$. Notice that $\deg(P_n) = n \cdot \mathcal{I}(n,q)$. Hence, $q^n = n \cdot \mathcal{I}(n,q) + \sum_{d|n, d<n} \deg(P_d)$, implying that $\mathcal{I}(n,q) \leq q^n/n$. This proves the upper bound.

Now, we will use this upper bound to prove the lower bound.

$$\begin{aligned}
q^n &= n \cdot \mathcal{I}(n,q) + \sum_{d|n, d<n} \deg(P_d) = n \cdot \mathcal{I}(n,q) + \sum_{d|n, d<n} d \cdot \mathcal{I}(d,q) \\
&\leq n \cdot \mathcal{I}(n,q) + \sum_{d \leq n/2} d \cdot \mathcal{I}(d,q) \\
&\leq n \cdot \mathcal{I}(n,q) + \sum_{d \leq n/2} q^d \quad \text{(using the upper bound)} \\
\Rightarrow n \cdot \mathcal{I}(n,q) &\geq q^n - \frac{q^{n/2+1} - 1}{q-1} \geq q^n - 2q^{n/2} \quad \text{(for } q > 1\text{)}
\end{aligned}$$

This proves the lower bound. ■

The total number of monic polynomials of degree n is q^n. Therefore, the density of monic irreducible polynomials of degree n is $\frac{\mathcal{I}(n,q)}{q^n}$, which lies between $1/2n$ and $1/n$, by the above lemma (assuming $q \geq 16$). This immediately suggests the following randomized algorithm.

Algorithm 1 Generating irreducible polynomial

1. Pick a monic polynomial f of degree n over $\mathbb{F}_q$, uniformly at random.
2. Test if f is irreducible.
3. If not, go to step 1.

Time complexity - Assuming $q \geq 16$, f is irreducible with probability at least $1/2n$. Therefore, after $2n$ iterations the probability that the algorithm hasn't found an irreducible polynomial is less than $1/e$. In other words, the expected number of iterations taken by the algorithm to pick an irreducible polynomial is $O(n)$.

This gives us a randomized procedure to generate irreducible polynomials. Unfortunately, there's no known deterministic polynomial time algorithm for generating irreducible polynomials. However, under the assumption of the Extended Riemann Hypothesis, such a deterministic algorithm exists [AJ86]. It is also known that the problem of generating an irreducible polynomial reduces in deterministic polynomial time to the problem of factoring polynomials over finite fields [Sho88]. No deterministic polynomial factoring algorithm is known even under the assumption of the Extended Riemann Hypothesis.

2 Miller-Rabin primality test

Distinguishing primes from composite numbers is one of the most fundamental problems in algorithmic number theory - this is known as the PRIMES problem. Till date, multiple randomized algorithms have been discovered to solve primality testing. In this section, we will discuss one such algorithm that is widely used in practice and is popularly known as the Miller-Rabin test [Mil76, Rab80]. It is a classic example of how randomization is used to design efficient algorithms in number theory.

Let $N > 0$ be an n-bit odd integer and $N - 1 = 2^t w$, where w is odd.

Algorithm 2 Miller-Rabin primality test

1. Choose a randomly from the range $[1, N-1]$.
2. If $\gcd(a, N) \neq 1$ return 'composite'.
3. If $a^{N-1} \neq 1 \mod N$ return 'composite'.
4. If $a^w = 1 \mod N$ return 'probably prime'.
5. Else, let $1 \leq r \leq t$ be the smallest possible integer such that $a^{2^r w} = 1 \mod N$.
6. If $a^{2^{r-1} w} \neq -1 \mod N$ return 'composite'. Else return 'probably prime'.

Correctness and success probability - First, it is easy to see that the algorithm always returns 'probably prime' if N is a prime. The reason being, if N is a prime then in step 3 $\gcd(a, N) = 1$ and hence from Fermat's little theorem $a^{N-1} = 1 \mod N$. Also in step 6, since $a^{2^r w} = 1 \mod N$ for the smallest possible $r \geq 1$, hence $a^{2^{r-1} w} = -1 \mod N$, $\mathbb{Z}_N$ being a field.

Let N be a composite. If the algorithm reaches step 3, we can assume that a has been chosen uniformly from $\mathbb{Z}_N^\times$, the set of positive integers coprime to and less than N. Now, if N is not a Carmichael number [1] then the set of integers a such that $a^{N-1} = 1 \mod N$ is a proper subgroup of $\mathbb{Z}_N^\times$ under multiplication modulo N (why?). Therefore, using Lagrange's theorem, the chance that $a^{N-1} \neq 1 \mod N$ is at least $\frac{1}{2}$.

Suppose N is a Carmichael number, which also means that N is square-free (why?). Without loss of generality assume that $N = pq$, where p and q are distinct primes. By the Chinese remaindering theorem, $\mathbb{Z}_N^\times \cong \mathbb{F}_p^\times \oplus \mathbb{F}_q^\times$. Let $p-1 = 2^k w_1$ and $q-1 = 2^\ell w_2$, where w_1 and w_2 are odd. And suppose $a = \beta^{s_1} \mod p =$

[1] A composite number N is called a Carmichael number, if $a^{N-1} = 1 \mod N$, for every integer a that is coprime to N.

$\gamma^{s_2} \bmod q$, where β and γ are generators of $\mathbb{F}_p^\times$ and $\mathbb{F}_q^\times$ respectively. In step 4, if $a^w = 1 \bmod N$ then $\beta^{s_1 w} = 1 \bmod p$ implying that $2^k | s_1$ as w is odd (why?). Similarly, $2^\ell | s_2$ if $a^w = 1 \bmod N$. Since a is randomly chosen from $\mathbb{Z}_N^\times$, equivalently s_1 and s_2 are chosen uniformly randomly and independently from the ranges $[1, p-1]$ and $[1, q-1]$ respectively. Therefore,

$$\begin{aligned} \Pr_a\{a^w = 1 \bmod N\} &\leq \Pr_{s_1,s_2}\{2^k | s_1 \text{ and } 2^\ell | s_2\} \\ &= \Pr_{s_1}\{2^k | s_1\} \cdot \Pr_{s_2}\{2^\ell | s_2\} = \frac{1}{2^{k+\ell}}. \text{ (why?)} \end{aligned}$$

Suppose in step 6, $a^{2^{r-1}w} = -1 \bmod N$. Then $\beta^{s_1 2^{r-1} w} = -1 \bmod p$ implying that 2^{k-r} *exactly* divides s_1 i.e. $k-r$ is the highest power of 2 that divides s_1 - we denote this exact division by $2^{k-r} \| s_1$. Similarly, $2^{\ell-r} \| s_2$. Notice that this also implies that $r \leq \min\{k, \ell\}$ (why?). For a fixed $r \leq \min\{k, \ell\}$,

$$\Pr_{s_1,s_2}\{2^{k-r} \| s_1 \text{ and } 2^{\ell-r} \| s_2\} = \frac{1}{2^{k+\ell-2(r-1)}}.$$

By union bound, over all $1 \leq r \leq \min\{k, \ell\} = k$ (say),

$$\Pr_a\{\exists r, 1 \leq r \leq t \text{ such that } a^{2^{r-1}w} = -1 \bmod N\} \leq \sum_{r=1}^{k} \frac{1}{2^{k+\ell-2(r-1)}}$$

Summing the error probabilities from step 4 and 6 we conclude that Miller-Rabin test succeeds with probability at least $1 - \frac{1}{2^{k+\ell}}(\frac{4^k+2}{3}) \geq \frac{1}{2}$.

Time complexity - Gcd computation in step 2 takes $O(\mathsf{M}_\mathsf{I}(n) \log n)$ time. In step 3 we can use repeated squaring and compute $a_1 = a^2 \bmod N$, $a_2 = a_1^2 = a^4 \bmod N$, $a_3 = a_2^2 = a^8 \bmod N$ and so on till $a_{\lfloor \log(N-1) \rfloor}$. Then, we can multiply all those a_i's modulo N for which the i^{th} bit of $N-1$ in binary is 1. This process takes $O(\mathsf{M}_\mathsf{I}(n) \log N) = O(n\mathsf{M}_\mathsf{I}(n))$ time. The complexity of steps 4, 5 and 6 are similarly bounded by $O(n\mathsf{M}_\mathsf{I}(n))$ as $r \leq t \leq \log N \leq n$. Therefore, the overall time complexity of the Miller-Rabin test is $O(n\mathsf{M}_\mathsf{I}(n)) = \tilde{O}(n^2)$ bit operations.

Remark - Another well known randomized primality test is the Solovay-Strassen test [SS77]. It is based upon the quadratic reciprocity theorem.

Deterministic primality test - In a major breakthrough, the first deterministic primality testing algorithm was given by Agrawal, Kayal and Saxena [AKS04] in 2002. It is famously known as the AKS primality test. The current best deterministic complexity is due to a version of the AKS-primality test given by Lenstra and Pomerance [JP05]. Their algorithm has a running time of $\tilde{O}(n^6)$ bit operations. We'll discuss this algorithm in the next few lectures.

Exercises:

1. Prove that a Carmichael number N is square-free, i.e. there's no prime p such that $p^2 | N$.
2. Show that 561 is a Carmichael number.
3. The analysis of the Miller-Rabin test is done by assuming that $N = p \cdot q$, where p and q are distinct primes. Show that a similar analysis holds for any general composite $N = p_1^{e_1} \dots p_r^{e_r}$, where $p_1, \dots, p_r$ are distinct primes.
4. Show that for any finite field $\mathbb{F}_q$, the multiplicative group $\mathbb{F}_q^\times = \mathbb{F}_q \backslash \{0\}$ is cyclic. Infer that, if β is a generator of $\mathbb{F}_q^\times$ and $\beta^k = 1$ in $\mathbb{F}_q$ then $q-1 | k$.

References

[AJ86] Leonard M. Adleman and Hendrik W. Lenstra Jr. Finding Irreducible Polynomials over Finite Fields. In *STOC*, pages 350–355, 1986.

[AKS04] Manindra Agrawal, Neeraj Kayal, and Nitin Saxena. Primes is in P. *Annals of Mathematics*, 160(2):781–793, 2004.

[JP05] Hendrik W. Lenstra Jr. and Carl Pomerance. Primality testing with Gaussian periods, July 2005. Available from `http://www.math.dartmouth.edu/ carlp/PDF/complexity12.pdf`.

[Mil76] Gary L. Miller. Riemann's hypothesis and tests for primality. *J. Comput. Syst. Sci.*, 13(3):300–317, 1976.

[Rab80] Michael O. Rabin. Probabilistic algorithm for testing primality. *J. Number Theory*, 12(1):128–138, 1980.

[Sho88] Victor Shoup. New Algorithms for Finding Irreducible Polynomials over Finite Fields. In *FOCS*, pages 283–290, 1988.

[SS77] Robert Solovay and Volker Strassen. A Fast Monte-Carlo Test for Primality. *SIAM J. Comput.*, 6(1):84–85, 1977.