PERIYAR UNIVERSITY

NAAC 'A++' Grade - State University
NIRF Rank 56 – State Public University Rank 25
SALEM - 636 011, Tamil Nadu, India.

CENTRE FOR DISTANCE AND ONLINE EDUCATION

(CDOE)

MASTER OF SCIENCE IN MATHEMATICS

SEMESTER - II

ELECTIVE COURSE: NUMBER THEORY AND CRYPTOGRAPHY

(Candidates admitted from 2024 onwards)

PERIYAR UNIVERSITY

CENTRE FOR DISTANCE AND ONLINE EDUCATION (CDOE)

M.Sc., MATHEMATICS 2024 admission onwards

ELECTIVE

Number Theory and Cryptography

Prepared by:

Centre for Distance and Online Education (CDOE)
Periyar University
Salem 636011

Contents

SYLLABUS: NUMBER THEORY AND CRYPTOGRAPHY

Objectives:

The objective of this course is to give elementary ideas from number theory which will have applications in cryptology.

Unit I: Elementary Number theory-I- Time estimates for doing arithmetic - divisibility and the Euclidean algorithm

Unit II: Elementary Number theory-II- Congruences - Some applications to factoring

Unit III: Finite Fields and Quadratic Residues - Finite Fields - Quadratic residues and reciprocity

Unit IV: Cryptography - Some simple cryptosystems - Enciphering matrices.

Unit V: Public Key Cryptography - Public key cryptography - RSA

References:

1. Neal Koblit, A course in Number Theory and Cryptography, Springer - Verlag, New York, 2nd edition, 2002.

Suggested Reading:

1. I. Niven and H. S. Zuckermann, An Introduction to Theory of Numbers (Edition 3), Wiley Eastern Ltd, New Delhi 1976

2. D. M. Burton, Elementary Number Theory, Brown Publishers, Iowa, 1989

3. K. Ireland and M. Rosen, A classic Introduction to Modern Number Theory, Springer - Verlag, 1972

4. N. Koblit, Algebraic Aspects of Cryptography, Springer-Verlag, 1998.

UNIT - 1

Unit 1

Elementary Number Theory-I

Objectives.

By studying this unit, the students will

1. recall the notation and facts from elementary number theory.
2. know the time estimates for doing arithmetic.
3. understand the big - O notation.
4. recall divisibility and the properties of divisibility.
5. know to apply the Euclidean algorithm for finding the $g.c.d.$ of two numbers.

1.1 Time estimates for doing arithmetic

Numbers in different bases: A non negative integer n written to the base b is a notation for n of the form $(d_{k-1}d_{k-2}\cdots d_1d_0)_b$, where the d's are digits, i.e., symbols for the integers between 0 and $b-1$; this notation

means that $n = d_{k-1}b^{k-1} + d_{k-2}b^{k-2} + \cdots + d_1 b + d_0$. If the first digit $d_k - 1$ is not zero, we call n a k-digit base-b number. Any number between b^{k-1} and b^k is a k-digit number to the base b. We shall omit the parentheses and subscript $(...)_b$ in the case of the usual decimal system ($b = 10$) and occasionally in other cases as well, if the choice of base is clear from the context, especially when we're using the binary system ($b = 2$). Since it is sometimes useful to work in bases other than 10, one should get used to doing arithmetic in an arbitrary base and to converting from one base to another. Fractions can also be expanded in any base, i.e., they can be represented in the form $(d_{k-l}d_{k-2}..d_1 d_0 d_{-l}d_{-2}...)_b$.

Remark 1.1.1. When $b > 10$ it is customary to use letters for the digits beyond 9. One could also use letters for all of the digits.

Example 1.1.2. 1. $(11001001)_2 = 201$.

2. When $b = 26$ let us use the letters $A - Z$ for the digits $0 - 25$, respectively. Then $(BAD)_{26} = 679$, whereas $(B.AD)_{26} = 1\frac{3}{676}$

Example 1.1.3. Multiply 160 and 199 in the base 7.

Solution.

$$\begin{array}{r} 3\ 1\ 6 \\ 4\ 0\ 3 \\ \hline 1\ 2\ 5\ 4 \\ 1\ 6\ 0\ 3\ 0 \\ \hline 1\ 6\ 1\ 5\ 5\ 4 \end{array}$$

Example 1.1.4. Divide $(11001001)_2$ by $(100111)_2$, and divide $(HAPPY)_{26}$ by $(SAD)_{26}$.

Solution.

$$\begin{array}{r} \quad\;\; 101\;\overline{110} \\ 100111\,\big)\overline{11001001} \\ \underline{100111} \\ \quad\;\; 101101 \\ \underline{100111} \\ \quad\;\; 110 \end{array} \qquad\qquad \begin{array}{r} \quad\;\; KD\;\overline{MLP} \\ SAD\,\big)\overline{HAPPY} \\ \underline{GYBE} \\ \quad\;\; COLY \\ \underline{CCAJ} \\ \quad\;\; MLP \end{array}$$

Example 1.1.5. Convert 10^6 to the bases 2, 7 and 26 (using the letters $A - Z$ as digits in the latter case).

Solution. To convert a number n to the base b, one first gets the last digit (the one's place) by dividing n by b and taking the remainder. Then replace n by the quotient and repeat the process to get the second-to-last digit d_1, and so on. Hence

$$10^6 = (11110100001001000000)_2 = (11333311)_7 = (CEXHO)_{26}.$$

Example 1.1.6. Convert $\pi = 3.1415926...$ to the base 2 (carrying out the computation 15 places to the right of the point) and to the base 26 (carrying out 3 places to the right of the point).

Solution. After taking care of the integer part, the fractional part is converted to the base b by multiplying by b, taking the integer part of the result as d_{-1}, then starting over again with the fractional part of what

you now have, successively finding $d_{-2}, d_{-3}, \cdots$. In this way one obtains:

$$3.1415926\cdots = (11.001001000011111\cdots)_2 = (D.DRS\cdots)_{26}.$$

Number of digits. As mentioned before, an integer n satisfying $b^{k-1} \leq n < b^k$ has k digits to the base b. By the definition of logarithms, this gives the following formula for the number of base-b digits (here "[ ]"denotes the greatest integer function):

$$\text{number of digits } = [\log_b n] + 1 = \left[\frac{\log n}{\log b}\right] + 1$$

where here (and from now on) "log"means the natural logarithm $\log_e$.

Bit operations. Let us start with a very simple arithmetic problem, the addition of two binary integers, for example:

$$\begin{array}{r} 1\ 1\ 1\ 1\ 0\ 0\ 0 \\ +\ 0\ 0\ 1\ 1\ 1\ 1\ 0 \\ \hline 1\ 0\ 0\ 1\ 0\ 1\ 1\ 0 \end{array}$$

Suppose that the number are both k bits long (the word "bit" is short for "binary digit"); if one of the two integers has fewer bits than the other, we fill in zero to the left, as in this example, to make them have the same length. Although this example involves small integers (adding 120 to 30), we should think of k as perhaps being very large, like 500 or 1000.

Remark 1.1.7. (**Procedure for adding k-bit numbers:**)

1. Look at the top and bottom bit, and also at whether there's a carry above the top bit.

2. If both bits are 0 and there is no carry, then put down 0 and move on.

3. If either (a) both bits are 0 and there is a carry, or (b) one of the bits is 0, the other is 1, and there is no carry, then put down 1 and move on.

4. If either (a) one of the bits is 0, the other is 1, and there is a carry, or else (b) both bits are 1 and there is no carry, then put down 0, put a carry in the next column, and move on.

5. If both bits are 1 and there is a carry, then put down 1, put a carry in the next column, and move on

Doing this procedure once is called a bit operation. Adding two k-bit numbers requires k bit operations.

Example 1.1.8. Find an upper bound for the number of bit operations required to compute $n!$.

Solution. We use the following procedure. First multiply 2 by 3, then the result by 4, then the result of that by $5, \cdots$, until you get to n. At the $(j-1)$-th step ($j = 2, 3, \cdots, n-1$), we are multiplying $j!$ by $j+1$. Hence we have $n-2$ steps, where each step involves multiplying a partial

product (*i.e.*, $j!$) by the next integer. The partial products will start to be very large. As a worst case estimate for the number of bits a partial product has, let's take the number of binary digits in the very last product, namely, in $n!$.

To find the number of bits in a product, we use the fact that the number of digits in the product of two numbers is either the sum of the number of digits in each factor or else 1 fewer than that sum. From this it follows that the product of n k-bit integers will have at most nk bits. Thus, if n is a k-bit integer - which implies that every integer less than n has at most k bits - then n! has at most nk bits.

Hence, in each of the $n-2$ multiplications needed to compute $n!$, we are multiplying an integer with at most k bits (namely $j+1$) by an integer with at most nk bits (namely $j!$). This requires at most nk^2 bit operations. We must do this $n-2$ times. So the total number of bit operations is bounded by $(n-2)nk^2 = n(n-2)([\log_2 n]+1)^2$. Roughly speaking, the bound is approximately $n^2(\log_2 n)^2$.

Example 1.1.9. Find an upper bound for the number of bit operations required to multiply a polynomial $\sum a_i x^i$ of degree $\leq n_1$ and a polynomial $\sum b_j x^j$ of degree $\leq n_2$ whose coefficients are positive integers $\leq m$. Suppose $n_2 \leq n_1$.

Solution. To compute $\sum_{i+j=v} a_i b_j$, which is the coefficient of x^v in the product polynomial (here $0 \leq v \leq n_l + n_2$) requires at most $n_2 + 1$ multi-

plications and n_2 additions. The numbers being multiplied are bounded by m, and the numbers being added are each at most m^2; but since we have to add the partial sum of up to n_2 such numbers we should take n_2m^2 as our bound on the size of the numbers being added.

Thus, in computing the coefficient of x^v the number of bit operations required is at most

$$(n_2 + 1)(log_2 m + 1)^2 + n_2(log_2(n_2 m^2) + 1).$$

Since there are $n_l + n_2 + 1$ values of v, our time estimate for the polynomial multiplication is

$$(n_l + n_2 + 1)((n_2 + 1)(log_2 m + 1)^2 + n_2(log_2(n_2 m^2) + 1)).$$

A slightly less rigorous bound is obtained by dropping the $1's$, there by obtaining an expression having a more compact appearance:

$$\frac{n_2(n_1 + n_2)}{log 2}\left(\frac{(log m)^2}{log 2} + (log\ n_2 + 2log\ m)\right).$$

Remark 1.1.10. If we set $n = n_l \geq n_2$ and make the assumption that $m > 16$ and $m \geq \sqrt{n_2}$ (which usually holds in practice), then the latter expression can be replaced by the much simpler $4n^2(\log_2 m)^2$.

Example 1.1.11. Find an upper bound for the number of bit operations it takes to compute the binomial coefficient $\binom{n}{m}$.

Solution. Since $\binom{n}{m} = \binom{n}{n-m}$, without loss of generality we may assume

that $m \leq n/2$. Let us use the following procedure to compute $\binom{n}{m} = n(n-l)(n-2)\cdots(n-m+1)/(2.3\cdots-m)$. We have $m-1$ multiplications followed by $m-1$ divisions. In each case the maximum possible size of the first number in the multiplication or division is $n(n-1)(n-2)\cdots(n-m+1) < n^m$, and a bound for the second number is n. Thus, we see that a bound for the total number of bit operations is $2(m-l)m([log\ 2n]+1)^2$, which for large m and n is essentially $2m^2(log_2 n)^2$. Notation for summarizing situation with time estimates:

The big-O notation. Suppose that $f(n)$ and $g(n)$ are functions of the positive integers n which take positive (but not necessarily integer) values for all n. We say that $f(n) = O(g(n))$ (or simply that $f = O(g)$) if there exists a constant C such that $f(n)$ is always less than $C.g(n)$. For example, $2n^2 + 3n - 3 = O(n^2)$ (namely, it is not hard to prove that the left side is always less than $3n^2$).

Because we want to use the big-O notation in more general situations, we shall give a more all-encompassing definition. Namely, we shall allow f and g to be functions of several variables, and we shall not be concerned about the relation between f and g for small values of n. Just as in the study of limits as $n \to \infty$ in calculus, here also we shall only be concerned with large values of n.

Definition 1.1.12. Let $f(n_1, n_2, \cdots, n_r)$ and $g(n_1, n_2, \cdots, n_r)$ be two functions whose domains are subsets of the set of all r-tuples of positive integers. Suppose that there exist constants B and C such that whenever

all of the n_j are greater than B the two functions are defined and positive, and $f(n_1, n_2, \cdots, n_r) < Cg(n_1, n_2, \cdots, n_r)$. In that case we say that f is bounded by g and we write $f = O(g)$.

Note that the "= "in the notation $f = O(g)$ should be thought of as more like a "<"and the big-O should be thought of as meaning "some constant multiple."

Example 1.1.13. 1. Let $f(n)$ be any polynomial of degree d whose leading coefficient is positive. Then it is easy to prove that $f(n) = O(nd)$. More generally, one can prove that $f = O(g)$ in any situation when $\dfrac{f(n)}{g(s)}$ has a finite limit as $n \to \infty$.

2. If ϵ is any positive number, no matter how small, then one can prove that $\log n = O(n^\epsilon)$ (i.e., for large 11, the log function is smaller than any power function, no matter how small the power). In fact. this follows because $\lim_{n\to\infty} \dfrac{\log n}{n^\epsilon}$, as one can prove using l'Hospital's rule.

3. If $f(n)$ denotes the number k of binary digits in n, then it follows from the above formulas for k that $f(n) = O(1ogn)$. Also notice that the same relation holds if $f(n)$ denotes the number of base-b digits, where b is any fixed base. On the other hand, suppose that the base b is not kept fixed but is allowed to increase, and we let $f(n, b)$ denote the number of base-b digits. Then we would want to use the relation $f(n, b) = O\left(\dfrac{\log n}{\log b}\right)$.

4. We have: Time $(n - m) = O(\log n. \log m)$, where the left hand side means the number of bit operations required to multiply n by m.

Illustration: In our use, the functions $f(n)$ or $f(n_1, n_2, \cdots, n_r)$ will often stand for the amount of time it takes to perform an arithmetic task with the integer n or with the set of integers $n_1, n_2, \cdots, n_r$ as input. We will want to obtain fairly simple-looking functions $g(n)$ as our bounds. When we do this, however, we do not want to obtain functions $g(n)$ which are much larger than necessary, since that would give an exaggerated impression of how long the task will take (although, from a strictly mathematical point of view, it is not incorrect to replace $g(n)$ by any larger function in the relation $f = O(g)$).

Roughly speaking, the relation $f(n) = O(n^d)$ tells us that the function f increases approximately like the d-th power of the variable.
For example, if $d = 3$, then it tells us that doubling n has the effect of increasing f by about a factor of 8. The relation $f(n) = O(\log^d n)$ (we write $\log^d n$ to mean $(\log n)^d$) tells us that the function increases approximately like the d-th power of the number of binary digits in n. That is because, up to a constant multiple, the number of bits is approximately $\log n$ (namely, it is within 1 of being $\dfrac{\log n}{\log 2} = 1.4427 \log n$). Thus, for example, if f $(n) = 0(\log^3 n)$, then doubling the number of bits in n (which is, of course, a much more drastic increase in the size of n than merely doubling n) has the effect of increasing f by about a factor of 8.

Note that to write $f(n) = O(1)$ means that the function f is bounded by some constant.

Remark 1.1.14. We have seen that, if we want to multiply two numbers

of about the same size, we can use the estimate $Time(k-bit.k-bit) = O(k^2)$. It should be noted that much work has been done on increasing the speed of multiplying two k-bit integers when k is large. Using clever techniques of multiplication that are much more complicated than the grade-school method we have been using, mathematicians have been able to find a procedure for multiplying two k-bit integers that requires only $O(k \log k \log\log k)$ bit operations. This is better than $O(k^2)$, and even better than $O(k^{1+\epsilon})$ for any $\epsilon > 0$, no matter how small. However, in what follows we shall always be content to use the rougher estimates above for the time needed for a multiplication.

In general, when estimating the number of bit operations required to do something, the first step is to decide upon and write down an outline of a detailed procedure for performing the task. An explicit step-by-step procedure for doing calculations is called an algorithm. Of course, there may be many different algorithms for doing the same thing. One may choose to use the one that is easiest to write down, or one may choose to use the fastest one known, or else one may choose to compromise and make a trade-off between simplicity and speed. The algorithm used above for multiplying n by m is far from the fastest one known. But it is certainly a lot faster than repeated addition (adding n to itself m times).

Example 1.1.15. Estimate the time required to convert a k-bit integer to its representation in the base 10.

Solution. Let n be a k-bit integer written in binary. The conversion

algorithm is as follows. Divide $10 = (1010)_2$ into n. The remainder - which will be one of the integers $0, 1, 10, 11, 100, 101, 110, 111, 1000$, or 1001 - will be the ones digit d_0. Now replace n by the quotient and repeat the process, dividing that quotient by $(1010)_2$, using the remainder as d_1 and the quotient as the next number into which to divide $(1010)_2$. This process must be repeated a number of times equal to the number of decimal digits in n, which is $\left[\frac{\log n}{\log 10}\right] + 1 = O(k)$. Then we're done. (We might want to take our list of decimal digits, i.e., of remainders from all the divisions, and convert them to the more familiar notation by replacing $0, 1, 10, 11, \cdots, 1001$ by $0, 1, 2, 3, \cdots, 9$, respectively.) How many bit operations does this all take? Well, we have $O(k)$ divisions, each requiring $O(4k)$ operations (dividing a number with at most k bits by the 4-bit number $(1010)_2$). But $O(4k)$ is the same as $O(k)$ (constant factors don't matter in the big-O notation), so we conclude that the total number of bit operations is $O(k).O(k) = O(k^2)$. If we want to express this in terms of n rather than k, then since $k = O(1ogn)$, we can write

$$\text{Time(convert } n \text{ to decimal)} \ = O(\log^2 n).$$

Example 1.1.16. Estimate the time required to convert a k-bit integer n to its representation in the base b, where b might be very large.

Solution. Using the same algorithm as in Example 10, except dividing now by the l-bit integer b, we find that each division now takes longer (if l is large), namely, $O(kl)$ bit operations. How many times do we have to

divide? Here notice that the number of base-b digits in n is $O\left(\frac{k}{l}\right)$. Thus, the total number of bit. operations required to do all of the necessary divisions is $O\left(\frac{k}{l}\right).O(kl) = O(k^2)$. Our estimate for the conversion time does not depend upon the base to which we're converting (no matter how large it may be). This is because the greater time required to find each digit is offset by the fact that there are fewer digits to be found.

Example 1.1.17. Express in terms of the O-notation the time required to compute (a) $n!$, (b) $\binom{n}{m}$ (see Examples 6 and 8).

Solution. (a) $O\left(n^2 \log^2 n\right)$ (b) $O\left(m^{2\log^2 n}\right)$.

In concluding this section, we make a definition that is fundamental in computer science and the theory of algorithms.

Definition 1.1.18. An algorithm to perform a computation involving integers $n_1, n_2, \cdots, n_r$ of $k_1, k_2, \cdots, k_r$ bits, respectively, is said to be a polynomial time algorithm if there exist integers $d_1, d_2, \cdots, d_r$ such that the number of bit operations required to perform the algorithm is $O\left(k_1^{d_1} k_2^{d_2} \cdots k_r^{d_r}\right)$.

Thus, the usual arithmetic operations $+, -, \times, \div$ are examples of polynomial time algorithms; so is conversion from one base to another. On the other hand, computation of $n!$ is not. (However, if one is satisfied with knowing $n!$ to only a certain number of significant figures, For example, its first 1000 binary digits, then one can obtain that by a polynomial time algorithm using Stirling's approximation formula for $n!$.)

Let Us Sum Up

- If n is a $k-$digit base-b number then $n = d_{k-1}b^{k-1} + d_{k-2}b^{k-2} + \cdots + d_1 b + d_0$, where $d_k - 1 \neq 0$.
- To convert a number n to the base b, first gets the one's place digit by dividing n by b and take the remainder.
- Adding two k-bit numbers requires k bit operations.
- To find the number of bits in a product, we use the fact that the number of digits in the product of two numbers is either the sum of the number of digits in each factor or else 1 fewer than that sum.
- We say that $f(n) = O(g(n))$ if there exists a constant C such that $f(n)$ is always less than $C.g(n)$.
- $f(n) = O(1)$ means that the function f is bounded by some constant.

Check your progress 1.1

1. Multiply $(212)_3$ by $(122)_3$.
2. Divide $(40122)_7$ by $(126)_7$.
3. Multiply the binary numbers 101101 and 11001, and divide 10011001 by 1011.
4. In the base 26, with digits A–Z representing 0-25, (a) multiply YES by NO, and (b) divide JQVXHJ by WE.

5. Write e = 2.7182818$\cdots$ (a) in binary 15 places out to the right of the point, and (b) to the base 26 out 3 places beyond the point.

1.2 Divisibility and the Euclidean algorithm

Divisors and divisibility. Given integers a and b, we say that a divides b (or "b is divisible by a") and we write $a|b$ if there exists an integer d such that $b = ad$. In that case we call a a divisor of b.

Every integer $b > 1$ has at least two positive divisors: 1 and b. By a proper divisor of b we mean a positive divisor not equal to b itself, and by a nontrivial divisor of b we mean a positive divisor not equal to 1 or b.

A prime number, by definition, is an integer greater than one which has no positive divisors other than 1 and itself.

A number is called composite if it has at least one nontrivial divisor.

The following properties of divisibility are easy to verify directly from the definition:

1. If $a|b$ and c is any integer, then $a|bc$.

2. If $a|b$ and $b|c$, then $a|c$.

3. If $a|b$ and $a|c$, then $a|b \pm c$.

   If p is a prime number and a is a nonnegative integer, then we use the notation $p^\alpha||b$ to mean that p^α is the highest power of p dividing b, i.e., that $p^\alpha|b$ and $p^{\alpha+1}$ does n.. In that case we say that p^α exactly divides b.

The Fundamental Theorem of Arithmetic states that any natural number n can be written uniquely as a product of prime numbers. It is customary to write this factorization as a product of distinct primes to the appropriate powers, listing the primes in increasing order. For example, $4200 = 2^3.3.5^2.7$.

Two consequences of the Fundamental Theorem are the following properties of divisibility:

4. If a prime number p divides ab, then either $p|a$ or $p|b$.

5. If $m|a$ and $n|a$, and if m and n have no divisors greater than 1 in common, then $mn|a$.

Another consequence of unique factorization is that it gives a systematic method for finding all divisors of n once n is written as a product of prime powers. Namely, any divisor d of n must be a product of the same primes raised to powers not exceeding the power that exactly divides n. That is, if $p^\alpha||n$, then $p^\beta||d$ for some β satisfying $\leq \beta \leq \alpha$. To find the divisors of 4200, for example, one takes 2 to the $0-$, $1-$, $2-$ or $3-$ power, multiplied by 3 to the $0-$ or $1-$ power, times 5 to the $0-$, $1-$ or $2-$ power, times 7 to the $0-$ or $1-$ power. The number of possible divisors is thus the product of the number of possibilities for each prime power, which, in turn, is $\alpha + 1$. That is, a number $n = p_1^{\alpha_1} p_2^{\alpha_2} \cdots p_r^{\alpha_r}$ has $(\alpha_1 + 1)(\alpha_2 + 1) \cdots (\alpha_r + 1)$ different divisors. For example, there are 48 divisors of 4200.

Given two integers a and b, not both zero, the greatest common divisor

of a and b, denoted $g.c.d.(a,b)$ (or sometimes simply (a,b)) is the largest integer d dividing both a and b. It is not hard to show that another equivalent definition of $g.c.d.(a,b)$ is the following: it is the only positive integer d which divides a and b and is divisible by any other number which divides both a and b.

If you happen to have the prime factorization of a and b in front of you, then it's very easy to write down $g.c.d.(a,b)$. Simply take all primes which occur in both factorizations raised to the minimum of the two exponents.

For example, comparing the factorization $10780 = 2^2.5.7^2.11$ with the above factorization of 4200, we see that $g.c.d.(4200, 10780) = 2^2.5.7 = 140$.

The least common multiple of a and b, denoted $l.c.m.(a,b)$. It is the smallest positive integer that both a and b divide. If you have the factorization of a and b, then you can get $l.c.m.(a,b)$ by taking all of the primes which occur in either factorization raised to the maximum of the exponents. It is easy to prove that $l.c.m.(a,b) = |ab|/g.c.d.(a,b)$.

The Euclidean algorithm. The greatest common divisor of two integers a and b can be found by listing all their positive divisors and choosing the largest one common to each: but this is cumbersome for large numbers. Fortunately, there's a relatively quick way to find $g.c.d.(a,b)$ even when you have no idea of the prime factors of a or b. It's called the Euclidean algorithm.

The Euclidean algorithm works as follows. To find $g.c.d.(a,b)$, where $a > b$, we first divide b into a and write down the quotient q_1 and the

remainder r_1 : $a = q_1 b + r_1$. Next, we perform a second division with b playing the role of a and r_1 playing the role of b : $b = q_2 r_1 + r_2$. Next, we divide r_2 into r_1 : $r_1 = q_3 r_2 + r_3$. We continue in this way, each time dividing the last remainder into the second-to-last remainder, obtaining a new quotient and remainder. When we finally obtain a remainder that divides the previous remainder, we are done: that final nonzero remainder is the greatest common divisor of a and b.

Example 1.2.1. Find $g.c.d.(1547, 560)$.

Solution.

$$1547 = 2 \cdot 560 + 427$$

$$560 = 1 \cdot 427 + 133$$

$$427 = 3 \cdot 133 + 28$$

$$133 = 4 \cdot 28 + 21$$

$$28 = 1 \cdot 21 + 7$$

Since $7|21$, we are done: $g.c.d.(1547, 560) = 7$.

Proposition 1.2.2. *The Euclidean algorithm always gives the greatest common divisor in a finite number of steps. In addition, for $a > b$*

$$\textit{Time (finding g.c.d.}(a, b) \textit{ by the Euclidean algorithm)} = O(\log^3(a)).$$

Proof. The proof of the first assertion is given in detail in many elementary number theory textbooks, so we merely summarize the argument.

First, it is easy to see that the remainders are strictly decreasing from one step to the next, and so must eventually reach zero. To see that the last remainder is the *g.c.d.*, use the second definition of the *g.c.d.* That is, if any number divides both a and b, it must divide r_l, and then, since it divides b and r_l, it must divide r_2, and so on, until you finally conclude that it must divide the last nonzero remainder. On the other hand, working from the last row up, one quickly sees that the last remainder must divide all of the previous remainders and also a and b. Thus, it is the *g.c.d.*, because the *g.c.d.* is the only number which divides both a and b and at the same time is divisible by any other number which divides a and b.

We next prove the time estimate. The main question that must be resolved is how many divisions we're performing. We claim that the remainders are not only decreasing, but they're decreasing rather rapidly. More precisely:

Claim. $r_{j+2} < \frac{1}{2} r_j$.

Proof of claim. First, if $r_{j+1} < \frac{1}{2} r_j$, then immediately we have $r_{j+2} < r_{j+1} < \frac{1}{2} r_j$. So suppose that $r_{j+1} > \frac{1}{2} r_j$. In that case the next division gives: $r_j = 1 \cdot r_{j+1} + r_{j+2}$, and so $r_{j+2} = r_j - r_{j+1} < \frac{1}{2} r_j$, as claimed.

We now return to the proof of the time estimate. Since every two steps must result in cutting the size of the remainder at least in half, and since the remainder never gets below 1, it follows that there are at most

$2 \cdot [\log_2 a]$ divisions. This is $O(\log a)$. Each division involves numbers no larger than a, and so takes $O(\log^2 a)$ bit operations. Thus, the total time required is $O(\log a) \cdot O(\log^2 a) = O(\log^3 a)$. This concludes the proof of the proposition. □

Remark 1.2.3. If one makes a more careful analysis of the number of bit operations, taking into account the decreasing size of the numbers in the successive divisions, one can improve the time estimate for the Euclidean algorithm to $O(\log^2 a)$.

Proposition 1.2.4. *Let $d = g.c.d.(a, b)$, where $a > b$. Then there exist integers u and v such that $d = ua + bv$. In other words, the g.c.d. of two numbers can be expressed as a linear combination of the numbers with integer coefficients. In addition, finding the integers u and v can be done in $O(\log^3 a)$ bit operations.*

Outline of proof. The procedure is to use the sequence of equalities in the Euclidean algorithm from the bottom up, at each stage writing d in terms of earlier and earlier remainders, until finally you get to a and b. At each stage you need a multiplication and an addition or subtraction. So it is easy to see that the number of bit operations is once again $O(\log^3 a)$.

Example 1.2.5. continued. To express 7 as a linear combination of 1547 and 560, we successively compute:

$$
\begin{aligned}
7 &= 28 - 1 \cdot 21 = 28 - 1(133 - 4 \cdot 28) \\
&= 5 \cdot 28 - 1 \cdot 133 = 5(427 - 3 \cdot 133) - 1 \cdot 133 \\
&= 5 \cdot 427 - 16 \cdot 133 = 5 \cdot 427 - 16(560 - 1 \cdot 427) \\
&= 21 \cdot 427 - 16 \cdot 560 = 21(1547 - 2 \cdot 560) - 16 \cdot 560 \\
&= 21 \cdot 1547 - 58 \cdot 560
\end{aligned}
$$

Definition 1.2.6. We say that two integers a and b are relatively prime (or that, "a is prime to b") if $g.c.d.(a, b) = 1$, i.e., if they have no common divisor greater than 1.

Corollary 1.2.7. *If $a > b$ are relatively prime integers, then 1 can be written as an integer linear combination of a and b in polynomial time, more precisely, in $O(log^3 a)$ bit operations.*

Definition 1.2.8. Let n be a positive integer. The Euler phi-function $\varphi(n)$ is defined to be the number of nonnegative integers b less than n which are prime to n:

$$\varphi(n)_{def} = |\{0 \leq b < n | g.c.d.(b, n) = 1\}| .$$

It is easy to see that $\varphi(1) = 1$ and that $\varphi(p) = p - 1$ for any prime p. We can also see that for any prime power

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

To see this, it suffices to note that the numbers from 0 to $p^\alpha - 1$ which are not prime to p^α are precisely those that are divisible by p, and there are

$p^{\alpha-1}$ of those. In the next section we shall show that the Euler φ-function has a "multiplicative property"that enables us to evaluate $\varphi(n)$ quickly, provided that we have the prime factorization of n. Namely, if n is written as a product of powers of distinct primes p^{α} then it turns out that $\varphi(n)$ is equal to the product of the $\varphi(p^{\alpha})$.

Let Us Sum Up

- A prime number is an integer greater than 1 which has no positive divisors other than 1 and itself.
- A number is called composite if it has at least one nontrivial divisor.
- By Fundamental Theorem of Arithmetic, any natural number n can be written uniquely as a product of prime numbers.
- $g.c.d.(a, b)$ is the largest integer d dividing both a and b.
- $l.c.m.(a, b)$ is the smallest positive integer that both a and b divide.
- The Euclidean algorithm always gives the greatest common divisor in a finite number of steps.
- The g.c.d. of two numbers can be expressed as a linear combination of the numbers with integer coefficients.
- If n is a positive integer then Euler phi-function $\varphi(n)$ is the number of nonnegative integers b less than n which are prime to n.
- If n is written as a product of powers of distinct primes p^{α} then it turns out that $\varphi(n)$ is equal to the product of the $\varphi(p^{\alpha})$.

Check your progress 1.2

1. How many divisors does 945 have? List them all.

2. Find $d = g.c.d.(360, 294)$ in two ways: (a) by finding the prime factorization of each number, and from that finding the prime factorization of d; and (b) by means of the Euclidean algorithm.

3. For each of the following pairs of integers, find their greatest common divisor using the Euclidean algorithm, and express it as an integer linear combination of the two numbers: (a) 26, 19; (b) 187, 34; (c) 841, 160; (d) 2613, 2171.

Unit Summary

In this unit we have discussed the time estimates for doing arithmetic, divisibility and the Euclidean algorithm. Also, we have studied how to apply the Euclidean algorithm to find the $g.c.d.$ of two numbers.

Glossary

Bit operation	- Binary digit operation.
Big-O	- Some constant multiple.
log	- Natural logarithm $\log_e$.
Prime number	- A positive integer divisible by only 1 and itself.
Composite number	- A number divisible by other numbers besides 1 and itself.
Prime factorization	- Expression of a number as a product of

prime numbers.

Relatively prime - Numbers with g.c.d. of 1.

Exercise 1.

1. By a "pure repeating" fraction of "period" f in the base b, we mean a number between 0 and 1 whose base-b digits to the right of the point repeat in blocks of f. For example, 1/3 is pure repeating of period 1 and 1/7 is pure repeating of period 6 in the decimal system. Prove that a fraction c/d (in lowest terms) between 0 and 1 is pure repeating of period f in the base b if and only if $b^f - 1$ is a multiple of d.

2. (a) The "hexadecimal" system means $b = 16$ with the letters A-F representing the tenth through fifteenth digits, respectively. Divide $(131B6C3)_{16}$ by $(1A2F)_{16}$.

   (b) Explain how to convert back and forth between binary and hexadecimal representations of an integer, and why the time required is far less than the general estimate given in Example 11 for converting from binary to base-b.

3. Describe a subtraction-type bit operation in the same way as was done for an addition-type bit operation in the text (the list of five alternatives).

4. (a) Using the big-O notation, estimate in terms of a simple function of n the number of bit operations required to compute 3^n in binary.

(b) Do the same for n?

5. Let n be a very large integer written in binary. Find a simple algorithm that computes $[\sqrt{n}]$ in $O(log^3 n)$ bit operations
(here [ ] denotes the greatest integer function)

6. Let n be a positive odd integer. (a) Prove that there is a 1-to-1 correspondence between the divisors of n which are $< \sqrt{n}$ and those that are $> \sqrt{n}$. (This part does not require n to be odd.)
(b) Prove that there is a 1-to-1 correspondence between all of the divisors of n which are $\geq \sqrt{n}$ and all the ways of writing n as a difference $s^2 - t^2$ of two squares nonnegative integers. (For example, 15 has two divisors 6, 15 that are $\geq \sqrt{15}$, and $15 = 4^2 - l^2 = 8^2 - 7^2$.)
(c) List all of the ways of writing 945 as a difference of two squares of nonnegative integers.

7. Find the power of each prime 2, 3, 5, 7 that exactly divides 100!, and then write out the entire prime factorization of 100!.

8. Suppose that a is much greater than b. Find a big$-O$ time estimate for $g.c.d.(a, b)$ that is better than $O(log^3 a)$.

9. The purpose of this problem is to find a "best possible" estimate for the number of divisions required in the Euclidean algorithm. The *Fibonacci numbers* can be defined by the rule $f_1 = 1$, $f_2 = 1$, $f_{n+1} = f_n + f_{n-1}$, for $n \geq 2$, or, equivalently, by means of the matrix equation

$$\begin{pmatrix} f_{n+1} & f_n \\ f_n & f_{n-1} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^n .$$

(a) Suppose that $a > b > 0$, and it takes k divisions to find $g.c.d.(a, b)$ by the Euclidean algorithm (the standard version given in the text, with nonnegative remainders). Show that $a \geq f_{k+2}$.

(b) Using the matrix definition of f_n, prove that

$f_n = \frac{\alpha^n - \alpha^m}{\sqrt{5}}$, where $\alpha = \frac{1+\sqrt{5}}{2}$, $\alpha' = \frac{1-\sqrt{5}}{2}$.

(c) Using parts (a) and (b), find an upper bound for k in terms of a. Compare with the estimate that follows from the proof of Proposition 1.2.2.

10. From algebra we know that a polynomial has a multiple root if and only if it has a common factor with its derivative; in that case the multiple roots of $f(x)$ are the roots of $g.c.d.(f, f')$. Find the multiple roots of the polynomial $x^4 - 2x^3 - x^2 + 2x + 1$.

Answers :

Check your progress 1.1

1. $(112111)_3$.
2. $(260\frac{12}{126})_7$.
3. 10001100101; 1101 $\frac{1010}{1011}$.
4. MPJNS; LIKE$\frac{\text{IT}}{\text{WE}}$(in other words, JQVXHJ=WE·LIKE + IT) .
5. (a) 10.101101111110000; (b) C.SR0.

Check your progress 1.2

1. 16 divisors: 1, 3, 5, 7, 9, 15, 21, 27, 35, 45, 63, 105, 135, 189, 315, 945.

3. (a) $1 = 11 \cdot 19 - 8 \cdot 26$; (b) $17 = 1 \cdot 187 - 5 \cdot 34$; (c) $1 = 205 \cdot 160 - 39 \cdot 841$; (d) $13 = 65 \cdot 2171 - 54 \cdot 2613$.

Exercise 1.

1. If $b^f - 1$ is a multiple of d, then the fraction can be written in the form $a/(b^f - 1)$, where a is an integer of at most f digits. Then use the formula for the sum of a geometric progression with initial term $a \cdot b^{-f}$ and ratio b^{-f}. Conversely, given a pure period $-f$ expansion x, you find that $b^f x$ differs from x by an f-digit integer a, and this means that $x = a/(b^f - 1)$.

2. (a) $(BAD)_{16}$; (b) no division is required: for example, to go from binary to hexadecimal simply start from the right and break off the digits in blocks of four; each four-tuple can be viewed as a hexadecimal digit (or replaced by one of the symbols 0-9, A-F).

3. (1) Look at the top and bottom bit and also at whether there's a borrow; (2) if both bits are the same and there is no borrow, or if the top bit is 1, the bottom bit is 0 and there is a borrow, then put down 0 and move on; (3) if the top bit is 1, the bottom bit is 0 and there is no borrow, then put down 1 and move on; (4) if the top bit is 0, the bottom bit is 1 and there is a borrow, then put down 0, put

a borrow in the next column, and move on; (5) if both bits are the same and there is a borrow, or if the top bit is 0, the bottom bit is 1 and there is no borrow, then put down 1, put a borrow in the next column, and move on.

4. (a) One needs $n-1$ multiplications; in each case the partial product 3^j has at most $O(n)$ digits and 3 has 2 digits, so there are $O(n)$ bit operations; thus, the total is $O(n^2) \cdot (b)$ Here the partial product has $O(n\ logn)$ digits, so each multiplication takes $O(n\ log^2 n)$ bit operations; the total is $O(n^2\ log^2 n)$.

5. Suppose that n has $k+1$ bits. As a first approximation to $m = [\sqrt{n}]$ take a 1 followed by $[k/2]$ zeros. Find the digits of m from left to right after the 1 by each time trying to change the zero to 1, and if the square of the resulting m is larger than n, putting it back to 0.

6. (a) When $a|n$ write $n = ab$ and let $a \mapsto b$. Given $n = ab$ with $a \geq b$, set $s = (a+b)/2$ and $t = (a-b)/2$. Conversely, given $n = s^2 - t^2$: set $a = s+t$, $b = s-t$ to get the reverse correspondence. (c) $473^2 - 472^2$, $159^2 - 156^2$, $97^2 - 92^2$, $71^2 - 64^2$, $57^2 - 48^2$, $39^2 - 24^2$, $33^2 - 12^2$, $31^2 - 4^2$.

7. $100! = 2^{97} \cdot 3^{48} \cdot 5^{24} \cdot 7^{16} \cdot 11^9 \cdot 13^7 \cdot 17^5 \cdot 19^5 \cdot 23^4 \cdot 29^3 \cdot 31^3 \cdot 31^3 \cdot 37^2 \cdot 41^2 \cdot 43^2 \cdot 53 \cdot 59 \cdot 61 \cdot 67 \cdot 71 \cdot 73 \cdot 79 \cdot 83 \cdot 89 \cdot 97$.

8. $O(log\ a\ log\ b\ + log^3\ b)$.

9. (a) The remainders decrease at the slowest rate when all of the quo-

tients are 1. (b) Write $\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} = BAB^{-1}$, where $A = \begin{pmatrix} \alpha & 0 \\ 0 & \alpha' \end{pmatrix}$ is the diagonal matrix made up from the eigenvalues and B is a matrix whose columns are eigenvectors, e.g., B $= \begin{pmatrix} \alpha & \alpha' \\ 1 & 1 \end{pmatrix}$.

(c) Since $\sqrt{5}a \geq \sqrt{5}f_{k+2} = \alpha^{k+2} - \alpha'^{k+2} > \alpha^{k+2} - 1$, it follows that $k < (log(1 + \sqrt{5}a)/log\alpha) - 2$; we can also get the simpler estimate $k < loga/log\alpha$. The latter estimate is equal to $1.44042 \cdots log_2 a$, while the estimate in the proof of Proposition 1.2.2 is $2log_2 a$.

10. $g.c.d.(f, f') = x^2 - x - 1$, and the multiple roots are the golden ratio and its conjugate $(1 \pm \sqrt{5})/2$.

References:

1. Neal Koblit, A course in Number Theory and Cryptography, Springer - Verlag, New York, 2nd edition, 2002.

Suggested Reading:

1. I. Niven and H. S. Zuckermann, An Introduction to Theory of Numbers (Edition 3), Wiley Eastern Ltd, New Delhi 1976

2. D. M. Burton, Elementary Number Theory, Brown Publishers, Iowa, 1989

3. K. Ireland and M. Rosen, A classic Introduction to Modern Number Theory, Springer - Verlag, 1972

4. N. Koblit, Algebraic Aspects of Cryptography, Springer-Verlag, 1998.

UNIT - 2

Unit 2

Elementary Number Theory-II

Objectives.

By studying this unit, the students will

1. understand the concept of congruences.

2. know to solve the problems using Chinese Remainder Theorem.

3. know to apply the concept of congruences to factoring.

2.1 Congruences

One of the most remarkable relations in number theory is the congruence relation, introduced and developed by the German mathematician Karl Friedrich Gauss, who is ranked with Archimedes (287-212 B.C.) and Issac Newton (1642-1727) as one of the greatest mathematicians of all time.

The congruence relation, as we will see shortly, shares many interesting

properties with the equality relation, so it is no accident that the congruence symbol $\equiv$, invented by Gauss around 1800, parallels the equality symbol $=$. The congruence symbol facilitates the study of divisibility theory and has many fascinating applications.

Definition 2.1.1. Given three integers a, b and m, we say that *"a is congruent to b modulo m"* and write $a \equiv b \mod m$, if the difference $a - b$ is divisible by m . m is called the *modulus* of the congruence.

Basic properties of congruences:

The following properties are easily proved directly from the definition:

1. (i)$a \equiv a \mod m$ (Reflexive Property);
   (ii)$a \equiv b \mod m$ if and only if $b \equiv a \mod m$ (Symmetric Property);
   (iii) if $a \equiv b \mod m$ and $b \equiv c \mod m$, then $a \equiv c \mod m$ (Transitive Property).
   For fixed m, (i)-(iii) mean that congruence modulo m is an **equivalence relation**.

2. For fixed m, each *equivalence class* with respect to congruence modulo m has one and only one representative between 0 and $m - 1$. In other words, any integer is congruent modulo m to one and only one integer between 0 and $m - 1$.
   The set of equivalence classes (called *residue classes*) will be denoted $\mathbb{Z}/m\mathbb{Z}$. Any set of representatives for the residue classes is called a **complete set of residues modulo m**.

3. If $a \equiv b \mod m$ and $c \equiv d \mod m$, then $a \pm c \equiv b \pm d \mod m$ and $ac \equiv bd \mod m$. In other words, congruences (with the same modulus) can be added, subtracted, or multiplied.

   One says that the set of equivalence classes $\mathbb{Z}/m\mathbb{Z}$ is a **commutative ring**, i.e., residue classes can be added, subtracted or multiplied (with the result not depending on which representatives of the equivalence classes were used), and these operations satisfy the familiar axioms (associativity, commutativity, additive inverse, etc.).

4. If $a \equiv b \mod m$, then $a \equiv b \mod d$ for any divisor $d|m$.

5. If $a \equiv b \mod m$, $a \equiv b \mod n$, and m and n are relatively prime, then $a \equiv b \mod mn$.

Proposition 2.1.2. *The elements of $\mathbb{Z}/m\mathbb{Z}$ which have multiplicative inverses are those which are relatively prime to m, i.e., the numbers a for which there exists b with $ab \equiv 1 \mod m$ are precisely those a for which g.c.d.(a,m)= 1. In addition, if g.c.d.(a,m) = 1, then such an inverse b can be found in $O(log^3 m)$ bit operations.*

Proof. First, if $d \equiv g.c.d.(a, m)$ were greater than 1, we could not have $ab \equiv 1 \mod m$ for any b, because that would imply that d divides $ab - 1$ and hence divides 1. Conversely, if $g.c.d.(a, m)$ =1, then by Property 2 of congruences we may suppose that $a < m$. Then , by Proposition 1.2.4, there exist integers u and v that can be found in $O(log^3 m)$ bit operations

for which $ua + vm = 1$. Choosing $b = u$, we see that $m|1 - ua = 1 - ab$, as desired.

Remark 2.1.3. If $g.c.d.(a, m) = 1$, then by negative powers $a^{-n} \mod m$ we mean the n-th power of the inverse residue class, i.e., it is represented by the n-th power of any integer b for which $ab \equiv 1 \mod m$.

Example 2.1.4. Find $160^{-1} \mod 841$, i.e., the inverse of 160 modulo 841.

Solution.

$$\begin{aligned}
841 &= 5 \cdot 160 + 41 \\
160 &= 3 \cdot 41 + 37 \\
41 &= 1 \cdot 37 + 4 \\
37 &= 9 \cdot 4 + 1 \\
g.c.d.of(160, 841) &= 1 \\
1 &= 37 - 9 \cdot 4 \\
&= 37 - 9 \cdot (41 - 1 \cdot 37) \\
&= 10 \cdot 37 - 9 \cdot 41 \\
&= 10 \cdot (160 - 3 \cdot 41) - 9 \cdot 41 \\
&= 10 \cdot 160 - 39 \cdot 41 \\
&= 10 \cdot 160 - 39 \cdot (841 - 5 \cdot 160) \\
&= 205 \cdot 160 - 39 \cdot 841
\end{aligned}$$

$$160^{-1} \mod 841 \equiv 205$$

Corollary 2.1.5. *If p is a prime number, then every nonzero residue class has a multiplicative inverse which can be found in $O(log^3 p)$ bit operations. We say that the ring $\mathbb{Z}/p\mathbb{Z}$ is a field. We denote this field $\boldsymbol{F}_p$, the "field of p elements."*

Corollary 2.1.6. *Suppose we want to solve a linear congruence $ax \equiv b \mod m$, where without loss of generality we may assume that $0 \leq a, b < m$. First, if g.c.d.$(a, m) = 1$, then there is a solution x_0 which can be found in $O(log^3 m)$ bit operations, and all solutions are of the form $x = x_0 + mn$ for n an integer.*

Next, suppose that $d =$ g.c.d.(a, m). There exists a solution if and only if $d|b$, and in that case our congruence is equivalent to the congruence $a'x \equiv b' \mod m'$, where $a' = a/d$, $b' = b/d$, $m' = m/d$.

Corollary 2.1.7. *If $a \equiv b \mod m$ and $c \equiv d \mod m$, and if g.c.d.(c, m) = 1 (in which case also g.c.d.(d,m)= 1), then $ac^{-1} \equiv bd^{-1} \mod m$ (where c^{-1} and d^{-1} denote any integers which are inverse to c and d modulo m).*

Proof. We have $c(ac^{-1} - bd^{-1}) \equiv (acc^{-1} - bdd^{-1}) \equiv a - b \equiv 0 \mod m$, and since m has no common factor with c, it follows that m must divide $ac^{-1} - bd^{-1}$.

Proposition 2.1.8. ***(Fermat's Little Theorem)*** *Let p be a prime. Any integer a satisfies $a^p \equiv a \mod p$, and any integer a not divisible by p satisfies $a^{p-1} \equiv 1 \mod p$.*

Proof. First suppose that $p \nmid a$. We first claim that the integers $0a, 1a, 2a, 3a, \cdots, (p-1)a$ are a complete set of residues modulo p. To

see this, we observe that otherwise two of them, say ia and ja, would have to be in the same residue class, i.e., $ia \equiv ja \mod p$. But this would mean that $p|(i-j)a$, and since a is not divisible by p, we would have $p|i-j$. Since i and j are both less than p, the only way this can happen is if $i = j$. We conclude that the integers $a, 2a, \cdots, (p-1)a$ are simply a rearrangement of $1, 2, \cdots, p-1$ when considered modulo p. Thus, it follows that the product of the numbers in the first sequence is congruent modulo p to the product of the numbers in the second sequence, i.e., $a^{p-1}(p-1)! \equiv (p-1)! \mod p$. Thus, $p|((p-1)!(a^{p-1}-1))$. Since $(p-1)!$ is not divisible by p, we have $p|(a^{p-1}-1)$, as required. Finally, if we multiply both sides of the congruence $a^{p-1} \equiv 1 \mod p$ by a, we get the first congruence in the statement of the proposition in the case when a is not divisible by p. But if a is divisible by p, then this congruence $a^p \equiv a \mod p$ is trivial, since both sides are $\equiv 0 \mod p$. This concludes the proof of the proposition.

Corollary 2.1.9. *If a is not divisible by p and if $n \equiv m \mod (p-1)$, then $a^n \equiv a^m \mod p$.*

Proof. Say $n > m$. Since $p-1|n-m$, we have $n = m + c(p-1)$ for some positive integer c. Then multiplying the congruence $a^{p-1} \equiv 1 \mod m$ by itself c times and then by $a^m \equiv a^m \mod p$ gives the desired result: $a^n \equiv a^m \mod p$.

Example 2.1.10. Find the last base-7 digit in $2^{1000000}$.

Solution. Let $p = 7$. Since 1000000 leaves a remainder of 4 when divided

by $p-1=6$, we have $2^{1000000} \equiv 2^4 = 16 \equiv 2 \mod 7$, so 2 is the answer.

Proposition 2.1.11. ***(Chinese Remainder Theorem).*** *Suppose that we want to solve a system of congruences to different moduli:*

$$\begin{aligned} x &\equiv a_1 \mod m_1, \\ x &\equiv a_2 \mod m_2, \\ &\cdots \cdots \\ x &\equiv a_r \mod m_r. \end{aligned}$$

Suppose that each pair of moduli is relatively prime: $g.c.d.(m_i, m_j) = 1$ for $i \neq j$. Then there exists a simultaneous solution x to all of the congruences, and any two solutions are congruent to one another modulo $M = m_1 m_2 \cdots m_r$.

Proof. First we prove uniqueness modulo M (the last sentence). Suppose that x' and x'' are two solutions. Let $x = x' - x''$. Then x must be congruent to 0 modulo each m_i, and hence modulo M (by Property 5 of congruences).

We next show how to construct a solution x.

Define $M_i = M/m_i$ to be the product of all the moduli *except* for the $i-th$. Clearly $g.c.d.(m_i, M_i) = 1$, and so there is an integer N_i (which can be found by means of the Euclidean algorithm) such that $M_i N_i \equiv 1 \mod m_i$. Now set $x = \sum_i a_i M_i N_i$. Then for each i we see that the terms in the sum other than the $i-th$ term are all divisible by m_i, because $m_i | M_j$ whenever $j \neq i$. Thus, for each i we have: $x \equiv a_i M_i N_i \equiv a_i \mod m_i$, as desired.

Corollary 2.1.12. *The Euler phi-function is "multiplicative", meaning that* $\varphi(mn) = \varphi(m)\varphi(n)$ *whenever* $g.c.d.(m,n) = 1$.

Proof. We must count the number of integers between 0 and $mn-1$ which have no common factor with mn. For each j in that range, let j_1 be its least nonnegative residue modulo m (i.e., $0 \leq j_1 < m$ and $j \equiv j_1 \mod m$) and let j_2 be its least nonnegative residue modulo n (i.e., $0 \leq j_2 < n$ and $j \equiv j_2 \mod n$). It follows from the Chinese Remainder Theorem that for each pair j_1,j_2 there is one and only one j between 0 and $mn-1$ for which $j \equiv j_1 \mod m$, $j \equiv j_2 \mod n$. Notice that j has no common factor with mn if and only if it has no common factor with m– which is equivalent to j_1 having no common factor with m–and it has no common factor with n– which is equivalent to j_2 having no common factor with n. Thus, the $j's$ which we must count are in 1-to-1 correspondence with the pairs j_1,j_2 for which $0 \leq j_1 < m$, $g.c.d.(j_1, m) = 1$; $0 \leq j_2 < n$, $g.c.d.(j_2, n) = 1$. The number of possible $j_1's$ is $\varphi(m)$, and the number of possible $j_2's$ is $\varphi(n)$. So the number of pairs is $\varphi(m)\varphi(n)$. This proves the corollary.

Remark 2.1.13. Since every positive integer n can be written as a product of prime powers, each of which has no common factors with the others , and since we know the formula $\varphi(p^\alpha) = p^\alpha(1 - \frac{1}{p})$, we can use the corollary to conclude that for $n = p_1^{\alpha_1} p_2^{\alpha_2} \cdots p_r^{\alpha_r}$: $\varphi(n) = p_1^{\alpha_1}(1 - \frac{1}{p_1}) p_2^{\alpha_2}(1 - \frac{1}{p_2}) \cdots p_r^{\alpha_r}(1 - \frac{1}{p_r}) = n \, \Pi_{p|n}(1 - \frac{1}{p})$.

Proposition 2.1.14. *Suppose that n is known to be the product of two distinct primes. Then knowledge of the two primes p, q is equivalent to knowledge of* $\varphi(n)$. *More precisely, one can compute* $\varphi(n)$ *from p,q in*

$O(logn)$ bit operations, and one can compute p and q from n and $\varphi(n)$ in $O(log^3 n)$ bit operations.

Proof. The proposition is trivial if n is even, because in that case we immediately know $p = 2$, $q = n/2$, and $\varphi(n) = n/2 - 1$; so we suppose that n is odd. By the multiplicativity of φ, for $n = pq$ we have $\varphi(n) = (p-1)(q-1) = n + 1 - (p+q)$. Thus $\varphi(n)$ can be found from p and q using one addition and one subtraction.

Conversely, suppose that we know n and $\varphi(n)$, but not p or q. We regard p, q as unknowns. We know their product n and also their sum, since $p+q = n+1-\varphi(n)$. Call the latter expression $2b$ (notice that it is even). But two numbers whose sum is $2b$ and whose product is n must be the roots of the quadratic equation $x^2 - 2bx + n = 0$. Thus, p and q equal $b \pm \sqrt{b^2 - n}$. This can be done in $O(log^3 n)$ bit operations. This completes the proof.

Proposition 2.1.15. ***(Euler's generalization of Fermat's Little Theorem)*** *If g.c.d.(a,m) = 1, then* $a^{\varphi(m)} \equiv 1 \mod m$.

Proof. We first prove the proposition in the case when m is a prime power: $m = p^\alpha$. We use induction on α. The case $\alpha = 1$ is precisely Fermat's Little Theorem. Suppose that $\alpha \geq 2$, and the formula holds for the $(\alpha-1)$-st power of p. Then $a^{p^{\alpha-1}-p^{\alpha-2}} = 1 + p^{\alpha-1}b$ for some integer b, by the induction assumption. Raising both sides of this equation to the p-th power and using the fact that the binomial coefficients in $(1+x)^p$ are each divisible by p (except in the 1 and x^p at the ends), we see that $a^{p^\alpha - p^{\alpha-1}}$

is equal to 1 plus a sum each term divisible by p^α. That is, $a^{\varphi(p^\alpha)} - 1$ is divisible by p^α, as desired. This proves the proposition for prime powers.

Finally, by the multiplicativity of φ, it is clear that $a^{\varphi(m)} \equiv 1 \mod p^\alpha$ (simply raise both sides of $a^{\varphi(p^\alpha)} \equiv 1 \mod p^\alpha$ to the appropriate power). Since this is true for each $p^\alpha || m$, and since the different prime powers have no common factors with one another, it follows by Property 5 of congruences that $a^{\varphi(m)} \equiv 1 \mod m$.

Corollary 2.1.16. *If g.c.d.(a,m) = 1 and if n' is the least nonnegative residue of n modulo $\varphi(m)$, then $a^n \equiv a^{n'} \mod m$.*

Remark 2.1.17. As the proof of Proposition 2.1.15 makes clear, there's a smaller power of a which is guaranteed to give 1 mod m: the least common multiple of the powers that give 1 mod p^α for each $p^\alpha || m$. For example, $a^{12} \equiv 1 \mod 105$ for a prime to 105, because 12 is a multiple of 3-1,5-1 and 7-1. Note that $\varphi(105) = 48$.

Example 2.1.18. Compute $2^{1000000} \mod 77$. **Solution.** Because 30 is

the least common multiple of $\varphi(7) = 6$ and $\varphi(11) = 10$, by the above Remark we have $2^{30} \equiv 1 \mod 77$. Since $1000000 = 30.33333 + 10$, it follows that $2^{1000000} \equiv 2^{10} \equiv 23 \mod 77$. A second method of solution would be first to compute $2^{1000000} \mod 7$ (since $1000000 = 6.166666 + 4$, this is $2^4 \equiv 2$) and also $2^{1000000} \mod 11$ (since 1000000 is divisible by 11-1, this is 1), and then use the Chinese Remainder Theorem to find an x between 0 and 76 which is $\equiv 2 \mod 7$ and $\equiv 1 \mod 11$.

Modular exponentiation by the repeated squaring method. A basic computation one often encounters in modular arithmetic is finding $b^n \mod m$ (i.e., finding the least nonnegative residue) when both m and n are very large. There is a clever way of doing this that is much quicker than repeated multiplication of b by itself. In what follows we shall assume that $b < m$, and that whenever we perform a multiplication we then immediately reduce $\mod m$ (i.e., replace the product by its least nonnegative residue). In that way we never encounter any integers greater than m^2.

We now describe the algorithm. Use a to denote the partial product. When we're done, we'll have a equal to the least nonnegative residue of $b^n \mod m$. We start out with $a = 1$. Let $n_0, n_1, \cdots, n_{k-1}$ denote the binary digits of n, i.e., $n = n_0 + 2n_1 + 4n_2 + \cdots + 2^{k-1}n_{k-1}$. Each n_j is 0 or 1. If $n_0 = 1$, change a to b (otherwise keep $a = 1$). Then square b, and set $b_1 = b^2 \mod m$ (i.e., b_1 is the least nonnegative residue of $b^2 \mod m$). If $n_1 = 1$, multiply a by b_1 (and reduce $\mod m$); otherwise keep a unchanged. Next square b_1, and set $b_2 = b_1^2 \mod m$. If $n_2 = 1$, multiply a by b_2; otherwise keep a unchanged. Continue in this way. You see that in the j-th step you have computed $b_j \equiv b^{2^j} \mod m$. If $n_j = 1$, i.e., if 2^j occurs in the binary expansion of n, then you include b_j in the product for a (if 2^j is absent from n, then you do not). It is easy to see that after the $(k-1)$-st step you'll have the desired $a \equiv b^n \mod m$.

Proposition 2.1.19. $Time(b^n \mod m) = O((logn)(log^2 m))$.

Remark 2.1.20. If n is very large in Proposition 2.1.19, you might want to use the Corollary of Proposition 2.1.15, replacing n by its least non-negative residue modulo $\varphi(m)$. But this requires that you know $\varphi(m)$. If you do know $\varphi(m)$, and if $g.c.d.(b, m) = 1$, so that you can replace n by its least nonnegative residue modulo $\varphi(m)$, then the estimate on the right in Proposition 2.1.19 can be replaced by $O(log^3 m)$.

Proposition 2.1.21. $\sum_{d|n} \varphi(d) = n$.

Proof. Let $f(n)$ denote the left side of equality in the proposition, i.e., $f(n)$ is the sum of $\varphi(d)$ taken over all divisors d of n (including 1 and n). We must show that $f(n) = n$. We first claim that $f(n)$ is multiplicative, i.e., that $f(mn) = f(m)f(n)$ whenever $g.c.d.(m, n) = 1$. To see this, we note that any divisor $d|mn$ can be written (in one and only one way) in the form $d_1 \cdot d_2$ where $d_1|m$, $d_2|n$. Since $g.c.d.(d_1, d_2) = 1$, we have $\varphi(d)$ $= \varphi(d_1)\varphi(d_2)$, because of the multiplicativity of φ. We get all possible divisors d of mn by taking all possible pairs d_1, d_2 where d_1 is a divisor of m and d_2 is a divisor of n. Thus, $f(mn) = \sum_{d_1|m} \sum_{d_2|n} \varphi(d_1)\varphi(d_2) =$ $\left(\sum_{d_1|m} \varphi(d_1)\right)\left(\sum_{d_2|n} \varphi(d_2)\right) = f(m)f(n)$, as claimed. Now to prove the proposition suppose that $n = p_1^{\alpha_1} \cdots p_r^{\alpha_r}$ is the prime factorization of n. By multiplicativity of f, we find that $f(n)$ is a product of terms of the form $f(p^\alpha)$. So it suffices to prove the proposition for p^α, i.e., to prove that $f(p^\alpha) = p^\alpha$. But the divisors of p^α are p^j for $0 \leq j \leq \alpha$, and so $f(p^\alpha)$ $= \sum_{j=0}^{\alpha} \varphi(p^j) = 1 + \sum_{j=1}^{\alpha}(p^j - p^{j-1}) = p^\alpha$. This proves the proposition for

p^α, and hence for all n.

Let Us Sum Up

- $a \equiv b \mod m$, if the difference $a - b$ is divisible by m.
- Congruence modulo m is an equivalence relation.
- The elements of $\mathbb{Z}/m\mathbb{Z}$ which have multiplicative inverses are those which are relatively prime to m.
- $a^{-n} \mod m$ means the inverse of a modulo m.
- Any integer $a^p \equiv a \mod p$, and any integer a not divisible by p satisfies $a^{p-1} \equiv \mod p$, where p is a prime.
- $\varphi(mn) = \varphi(m)\varphi(n)$ whenever $g.c.d.(m, n) = 1$.
- If g.c.d.(a,m) = 1, then $a^{\varphi(m)} \equiv 1 \mod m$.
- $\sum_{d|n} \varphi(d) = n$.

Check your progress 2.1

1. Describe all of the solutions of the following congruences:

   (a) $3x \equiv 4 \mod 7$; (d) $27x \equiv 25 \mod 256$;

   (b) $3x \equiv 4 \mod 12$; (e) $27x \equiv 72 \mod 900$;

   (c) $9x \equiv 12 \mod 21$; (f) $103x \equiv 612 \mod 676$.

2. Find a 3-digit (decimal) number which leaves a remainder of 4 when divided by 7, 9, or 11.

3. Find the smallest positive integer which leaves a remainder of 1 when divided by 11, a remainder of 2 when divided by 12, and a remainder of 3 when divided by 13.

4. Use the repeated squaring method to find $38^{75} \mod 103$.

5. Find $\varphi(n)$ for all m from 90 to 100.

2.2 Some applications to factoring

Proposition 2.2.1. *For any integer b and any positive integer n, $b^n - 1$ is divisible by $b-1$ with quotient $b^{n-1} + b^{n-2} + \cdots + b^2 + b + 1$.*

Proof. We have a polynomial identity coming from the following fact: 1 is a root of $x^n - 1$, and so the linear term $x - 1$ must divide $x^n - 1$. Namely, polynomial division gives $x^n - 1 = (x-1)(x^{n-1} + x^{n-2} + \cdots + x^2 + x + 1)$. (Alternatively, we can derive this by multiplying x by $x^{n-1} + x^{n-2} + \cdots + x^2 + x + 1$, then subtracting $x^{n-1} + x^{n-2} + \cdots + x^2 + x + 1$, and finally obtaining $x^n - 1$ after all the canceling.) Now we get the proposition by replacing x by b.

A second proof is to use arithmetic in the base b. Written to the base b, the number $b^n - 1$ consists of n digits $b - 1$ (for example, $10^6 - 1 = 999999$). On the other hand, $b^{n-1} + b^{n-2} + \cdots + b^2 + b + 1$ consists of n digits all 1. Multiplying $111\cdots111$ by the 1-digit number $b - 1$ gives $(b-1)(b-1)(b-1)\cdots(b-1)(b-1)(b-1)_b = b^n - 1$.

Corollary 2.2.2. *For any integer b and any positive integers m and n,*

we have, $b^{mn} - 1 = (b^m - 1)(b^{m(n-1)} + b^{m(n-2)} + \cdots + b^{2m} + b^m + 1)$.

Proof. Simply replace b by b^m in the last proposition.

Example 2.2.3. Using the above corollary, we see that $2^{35} - 1$ is divisible by $2^5 - 1 = 31$ and by $2^7 - 1 = 127$. Namely, we set $b = 2$ and either $m = 5$, $n = 7$ or else $m = 7$, $n = 5$.

Proposition 2.2.4. *Suppose that b is prime to m, and a and c are positive integers. If* $b^a \equiv 1 \mod m$ *and* $b^c \equiv 1 \mod m$, *and if* $d = g.c.d.(a, c)$, *then* $b^d \equiv 1 \mod m$.

Proof. Using the Euclidean algorithm, we can write d in the form $ua+vc$, where u and v are integers. It is easy to see that one of the two numbers u, v is positive and the other is negative or zero. Without loss of generality, we may suppose that $u > 0$, $v \leq 0$. Now raise both sides of the congruence $b^a \equiv 1 \mod m$ to the u-th power, and raise both sides of the congruence $b^c \equiv 1 \mod m$ to the $(-v)$-th power. Now divide the resulting two congruences, obtaining: $b^{au-c(-v)} \equiv 1 \mod m$. But $au + cv = d$, so the proposition is proved.

Proposition 2.2.5. *If p is a prime dividing* $b^n - 1$, *then either (i)* $p | b^d - 1$ *for some proper divisor d of n, or else (ii)* $p \equiv 1 \mod n$. *If* $p > 2$ *and n is odd, then in case (ii) one has* $p \equiv 1 \mod 2n$.

Proof. We have $b^n \equiv 1 \mod p$ and also, by Fermat's Little Theorem, we have $b^{p-1} \equiv 1 \mod p$. By the above proposition, this means that $b^d \equiv 1 \mod p$, where $d = g.c.d.(n, p - 1)$. First, if $d < n$, then this says that

$p|b^d-1$ for a proper divisor d of n, i.e., case (i) holds. On the other hand, if $d=n$, then, since $d|p-1$, we have $p\equiv 1 \mod n$. Finally, if p and n are both odd and $n|p-1$ (i.e., we're in case (ii)), then obviously $2n|p-1$.

Example 2.2.6. Factor $2^{11}-1=2047$.

Solution. If $p|2^{11}-1$, by the theorem we must have $p\equiv 1 \mod 22$. Thus, we test $p=23,67,89,\cdots$ (actually, we need go no farther that $\sqrt{2047}=45.\cdots$). We immediately obtain the prime factorization of 2047: $2047=23\cdot 89$. In a very similar way, one can quickly show that $2^{13}-1=8191$ is prime. A prime of the form 2^n-1 is called a "Mersenne prime".

Example 2.2.7. Factor $3^{12}-1=531440$.

Solution. By the proposition above, we first try the factors of the much smaller numbers 3^1-1, 3^2-1, 3^3-1, 3^4-1, and the factors of 3^6-1 $=(3^3-1)(3^3+1)$ which do not already occur in 3^3-1. This gives us $2^4\cdot 5\cdot 7\cdot 13$. Since $531440/(2^4\cdot 5\cdot 7\cdot 13)=73$, which is prime, we are done. Note that, as expected, any prime that did not occur in 3^d-1 for d a proper divisor of 12–namely, 73 must be $\equiv 1 \mod 12$.

Example 2.2.8. Factor $2^{35}-1=34359738367$.

Solution. First we consider the factors of 2^d-1 for $d=1$, 5, 7. This gives the prime factors 31 and 127. Now $(2^{35}-1)/(31\cdot 127)=8727391$. According to the proposition, any remaining prime factor must be $\equiv 1$ mod 70. So we check 71, 211, 281,$\cdots$, looking for divisors of 8727391. At first, we might be afraid that we'll have to check all such prime less that $\sqrt{8727391}=2954.\cdots$. However, we immediately find that $8727391=71\cdot$

122921, and then it remains to check only up to $\sqrt{122921} = 350.\cdots$. We find that 122921 is prime. Thus, $2^{35}-1 = 31\cdot 71\cdot 127\cdot 122921$ is the prime factorization.

Remark 2.2.9. In Example 2.2.8, how can one do the arithmetic on a calculator that only shows, say, 8 decimal places? Simply break up the numbers into sections. For example, when we compute 2^{35}, we reach the limit of our calculator display with $2^{26} = 67108864$. To multiply this by $2^9 = 512$, we write $2^{35} = 512\cdot (67108\cdot 1000 + 864) = 34359296\cdot 1000 + 442368 = 34359738368$. Later, when we divide $2^{35}-1$ by $31\cdot 127 = 3937$, we first divide 3937 into 34359738, taking the integer part of the quotient: $\left[\frac{34359738}{3937}\right] = 8727$. Next, we write $34359738 = 3937\cdot 8727 + 1539$. Then

$$\begin{aligned}\frac{34359738367}{3937} &= \frac{(3937\cdot 8727+1539)\cdot 1000+367}{3937}\\ &= 8727000 + \frac{1539367}{3937}\\ &= 8727391.\end{aligned}$$

Remark 2.2.10. A prime number of the form 2^n-1 is called a "Mersenne Prime" and a prime number of the form 2^n+1 is called a "Fermat Prime".

The first few Mersenne primes are 3,7,31,121 and the first few Fermat primes are 3,5,17,257.

Let Us Sum Up

- For any integer b and any positive integer n, $b^n - 1$ is divisible by $b-1$ with quotient $b^{n-1} + b^{n-2} + \cdots + b^2 + b + 1$.

- For any integer b and any positive integers m and n, we have, $b^{mn}-1 = (b^m-1)(b^{m(n-1)}+b^{m(n-2)}+\cdots+b^{2m}+b^m+1)$.
- If $b^a \equiv 1 \mod m$, $b^c \equiv 1 \mod m$, $d = g.c.d.(a,c)$ where b is prime to m and a, c are positive integers, then $b^d \equiv 1 \mod m$.
- If p is a prime dividing b^n-1, then either $p|b^d-1$ for some proper divisor d of n or $p \equiv 1 \mod n$. If $p > 2$ and n is odd, then $p \equiv 1 \mod 2n$.
- A prime number of the form 2^n-1 is a Mersenne Prime.
- A prime number of the form 2^n+1 is a Fermat Prime.

Check your progress 2.2

1. Factor $3^{15}-1$ and $3^{24}-1$.
2. Factor $5^{12}-1$.
3. Factor 10^5-1, 10^6-1 and 10^8-1.
4. Factor $2^{33}-1$ and $2^{21}-1$.
5. Factor $2^{15}-1$, $2^{30}-1$, and $2^{60}-1$.

Unit Summary

In this unit we have discussed the basic properties of congruences, Fermat's Little Theorem, Chinese Remainder Theorem and Euler's generalization of Fermat's Little Theorem. Also, we studied the method of

solving problems of congruences using repeating square method and some applications of congruences to factoring.

Glossary

Equivalence relation	- A relation that is reflexive, symmetric and transitive.
Equivalence class	- A set of elements equivalent to each other.
Residue class	- A set of equivalence classes.
Commutative ring	- A ring where multiplication operation is commutative.
Mersenne prime	- Prime number of the form $2^n - 1$.
Fermat prime	- Prime number of the form $2^n + 1$.

Exercise 2.

1. What are the possibilities for the last hexadecimal digit of a perfect square? (see 2nd Problem of Exercise 1).

2. Prove that $n^5 - n$ is always divisible by 30.

3. Find the smallest nonnegative solution of each of the following system of congruences:

   (a) $x \equiv 2 \mod 3$
   $x \equiv 3 \mod 5$
   $x \equiv 4 \mod 11$
   $x \equiv 5 \mod 16$

   (b) $x \equiv 12 \mod 31$
   $x \equiv 87 \mod 127$
   $x \equiv 91 \mod 255$

   (c) $19x \equiv 103 \mod 900$
   $10x \equiv 511 \mod 841$

4. Suppose that a 3-digit (decimal) positive integer which leaves a remainder of 7 when divided by 9 or 10 and 3 when divided by 11 goes evenly into a six-digit natural number which leaves a remainder of 8 when divided by 9, 7 when divided by 10, and 1 when divided by 11. Find the quotient.

5. Make a list showing all n for which $\varphi(n) \leq 12$, and prove that your list is complete.

6. Suppose that n is not a perfect square, and that $n - 1 > \varphi(n) > n - n^{2/3}$. Prove that n is a product of two distinct primes.

7. Suppose that b is prime to m , where $m > 2$, and a and c are positive integers. Prove that, if $b^a \equiv -1 \mod m$ and $b^c \equiv \pm 1 \mod m$, and if $d = g.c.d.(a, c)$, then $b^d \equiv -1 \mod m$, and a/d is odd.

8. Let $m = 2^{24} + 1 = 16777217$.
   (a) Find a Fermat prime which divides m.
   (b) Find the complete prime factorization of m.

9. Prove that if $d = g.c.d.(m, n)$ and $a > 1$ is an integer, then $g.c.d.(a^m - 1, a^n - 1) = ad - 1$.

Answers.

Check your progress 2.1

1. (a) $x = 6 + 7n$, n any integer; (b) no solution; (c) same as (a); (d) $219 + 256n$; (e) $36 + 100n$; (f) $636 + 676n$.

2. Of course, 4 has the desired property, but it is not a 3-digit number. By the last part of the Chinese Remainder Theorem, any other number which leaves the right remainders must differ from 4 by a multiple of $7 \cdot 9 \cdot 11 = 693$. The only 3-digit possibility is $4 + 693 = 697$.

3. One can apply the Chinese Remainder Theorem to the congruences $x \equiv 1 \mod 11$, $x \equiv 2 \mod 12$, $x \equiv 3 \mod 13$. Alternately, one can observe that obviously -10 leaves the right remainders, and then get $-10 + 11 \cdot 12 \cdot 13 = 1706$.

4. $38^{1+2+2^3+2^6} = 38 \cdot 2 \cdot 16 \cdot 63 = 79 \mod 103$.

5.

n	90	91	92	93	94	95	96	97	98	99	100
$\varphi(n)$	24	72	44	60	46	72	32	96	42	60	40

Check your progress 2.2

1. $2 \cdot 11^2 \cdot 13 \cdot 4561$, $2^5 \cdot 5 \cdot 7 \cdot 13 \cdot 41 \cdot 73 \cdot 6481$.

2. $2^4 \cdot 3^2 \cdot 7 \cdot 13 \cdot 31 \cdot 601$.

3. $3^2 \cdot 41 \cdot 271$, $3^3 \cdot 7 \cdot 11 \cdot 13 \cdot 37$, $3^2 \cdot 11 \cdot 73 \cdot 101 \cdot 137$.

4. $7 \cdot 23 \cdot 89 \cdot 599479$; $7^2 \cdot 127 \cdot 337$ (this example shows that a prime $p|b^d - 1$ in Proposition 2.2.5 may divide $b^n - 1$ to a greater power than it divides $b^d - 1$).

5. $7\cdot\ 31\cdot\ 151$, $3^2\cdot\ 7\cdot\ 11\cdot\ 31\cdot\ 151\cdot\ 331$, $3^2\cdot\ 5^2\cdot\ 7\cdot\ 11\cdot\ 13\cdot\ 31\cdot\ 41\cdot\ 61\cdot\ 151\cdot\ 331\cdot 1321$.

Exercise 2.

1. 0, 1, 4, 9.

2. Prove separately that it is divisible by 2, 3 and 5.

3. (a) 1973; (b) 63841; (c) 58837.

4. The quotient leaves remainders of 5, 1, 4 when divided by 9, 10, 11, and so (by the Chinese Remainder Theorem) is of the form $851 + 990m$. Similarly, the divisor is of the form $817 + 990n$. Since the divisor has 3 digits, $n = 0$. Since the product has 6 digits, also $m = 0$. Thus, the answer is 851.

5. There is no n for which $\varphi(n)$ is an odd number greater than 1; $\varphi(n) = 1$ for $n = 1, 2$; $\varphi(n) = 2$ for $n = 3, 4, 6$; $\varphi(n) = 4$ for $n = 5, 8, 10, 12$; $\varphi(n) = 6$ for $n = 7, 9, 14, 18$; $\varphi(n) = 8$ for $n = 15, 16, 20, 24, 30$; $\varphi(n) = 10$ for $n = 11, 22$; $\varphi(n) = 12$ for $n = 13, 21, 26, 28, 36, 42$. To prove, for example, that these are all of the n for which $\varphi(n) = 12$, compare the possible factorizations of 12 (with 1 allowed as a factor but not 3) with the formula $\varphi(\prod p^\alpha) = \prod(p^\alpha - p^{\alpha-1})$. One has $1\cdot 2\cdot 6$, $1\cdot 12$, $2\cdot 6$, and 12. The first gives $2\cdot 3\cdot 7$, the second gives $2\cdot 13$, the third gives (3 or 4) $\cdot$ 7 and $4\cdot 9$, and the fourth gives 13.

6. n cannot be a prime, since if it were $\varphi(n) = n-1$. By assumption, n is not the square of a prime. If it were not a product of two distinct

primes, then it would be a product of three or more primes (not necessarily distinct). Let p be the smallest. Then $p \leq n^{1/3}$, and we have $\varphi(n) \leq n(1 - \frac{1}{p}) \leq n(1 - n^{-1/3}) = n - n^{2/3}$, a contradiction.

7. Use the same argument as in the proof of the last proposition to conclude that $b^d \equiv \pm 1 \mod m$. But since $(b^d)^{a/d} \equiv -1 \mod m$, it follows that $b^d \equiv -1 \mod m$ and a/d is odd.

8. (a) $2^8 + 1 = 257$; (b) $m = 97 \cdot 257 \cdot 673$.

9. Apply side by side the Euclidean algorithm to find $g.c.d.(a^m - 1, a^n - 1)$ and to find $g.c.d.(m, n)$. Notice that at each stage the remainder in the first Euclidean algorithm is $a^r - 1$, where r is the remainder in the second Euclidean algorithm. For example, in the first step one divides $a^m - 1$ by $a^n - 1$ to get $a^r - 1$, where r is the remainder when m is divided by n.

References:

1. Neal Koblit, A course in Number Theory and Cryptography, Springer - Verlag, New York, 2nd edition, 2002.

Suggested Reading:

1. I. Niven and H. S. Zuckermann, An Introduction to Theory of Numbers (Edition 3), Wiley Eastern Ltd, New Delhi 1976

2. D. M. Burton, Elementary Number Theory, Brown Publishers, Iowa, 1989

3. K. Ireland and M. Rosen, A classic Introduction to Modern Number Theory, Springer - Verlag, 1972

4. N. Koblit, Algebraic Aspects of Cryptography, Springer-Verlag, 1998.

123

UNIT - 3

Unit 3

Finite Fields and Quadratic Residues

Objectives.

By studying this unit, the students will

1. recall the basic definitions and properties of a field.
2. know about finite fields.
3. understand the existence of multiplicative generators of finite field.
4. know to find the quadratic residues and reciprocity.
5. know how to find the square root of a residue.
6. understand the Legendre symbol and its properties.

3.1 Basic definitions and Properties of a field.

Definition 3.1.1. A field is a set **F** with a *multiplication* and *addition* operation which satisfy the familiar rules– associativity and commutativity of both addition and multiplication, the distributive law, existence of

an additive identity 0 and a multiplicative identity 1, additive inverses, and multiplicative inverses for everything except 0.

Example 3.1.2. The following are the examples of field:

(1) the field $\mathbf{Q}$ consisting of all rational numbers;

(2) the field $\mathbf{R}$ of real numbers;

(3) the field $\mathbf{C}$ of complex numbers;

(4) the field $\mathbb{Z}/p\mathbb{Z}$ of integers modulo a prime number p.

Definition 3.1.3. A **vector space** can be defined over any field $\mathbf{F}$ by the same properties that are used to define a vector space over the real numbers. Any vector space has a **basis**, and the number of elements in a basis is called its **dimension**. An **extension field**, i.e., a bigger field containing $\mathbf{F}$, is automatically a vector space over $\mathbf{F}$. We call it a **finite extension** if it is a finite dimensional vector space. By the **degree** of a finite extension we mean its dimension as a vector space. One common way of obtaining extension fields is to **adjoin** an element to $\mathbf{F}$: we say that $\mathbf{K} = \mathbf{F}(\alpha)$ if $\mathbf{K}$ is the field consisting of all rational expressions formed using α and elements of $\mathbf{F}$.

Definition 3.1.4. The **polynomial ring** can be defined over any field $\mathbf{F}$. It is denoted $\mathbf{F}[X]$; it consists of all finite sums of powers of X with coefficients in $\mathbf{F}$. One adds and multiplies polynomials in $\mathbf{F}[X]$ in the same way as one does with polynomials over the reals. The **degree** d of a polynomial is the largest power of X which occurs with nonzero

coefficient; in a **monic** polynomial the coefficient of X^d is 1. We say that g **divides** f, where $f, g \in \mathbf{F}[X]$, if there exists a polynomial $h \in \mathbf{F}[X]$ such that $f = gh$. The irreducible polynomials $f \in \mathbf{F}[X]$ are those that are not divisible by any polynomials of lower degree except for constants; they play the role among the polynomials that the primes play among the integers. The polynomial ring has **unique factorization**, meaning that every monic polynomial can be written in one and only one way (except for the order of factors) as a product of monic irreducible polynomials.(A non-monic polynomial can be uniquely written as a constant times such a product.)

Definition 3.1.5. An element α in some extension field $\mathbf{K}$ containing $\mathbf{F}$ is said to be **algebraic** over $\mathbf{F}$ if it satisfies a polynomial with coefficients in $\mathbf{F}$. In that case there is a **unique** monic irreducible polynomial in $\mathbf{F}[X]$ of which α is a root (and any other polynomial which α satisfies must be divisible by this monic irreducible polynomial). If this monic irreducible polynomial has degree d, then any element of $\mathbf{F}(\alpha)$ (i.e., any rational expression involving powers of α and elements in $\mathbf{F}$) can actually be expressed as a linear combination of the powers 1, α, α^2,$\cdots$,$\alpha^{(d-1)}$. Thus, those powers of α form a basis of F(α) over F, and so the degree of the extension obtained by adjoining α is the same as the degree of the monic irreducible polynomial of α. Any other root α' of the same irreducible polynomial is called a **conjugate** of α over $\mathbf{F}$. The fields $\mathbf{F}(\alpha)$ and $\mathbf{F}(\alpha')$ are **isomorphic** by means of the map that takes any expression in terms of α to the same expression with α replaced by α'. The word

"isomorphic" means that we have a 1-to-1 correspondence that preserves addition and multiplication. In some cases the fields $\mathbf{F}(\alpha)$ and $\mathbf{F}(\alpha')$ are the same, in which case we obtain an **automorphism** of the field. For example, $\sqrt{2}$ has one conjugate, namely $-\sqrt{2}$, over $\mathbf{Q}$, and the map $a+b\sqrt{2} \mapsto a-b\sqrt{2}$ is an automorphism of the field $\mathbf{Q}(\sqrt{2})$ (which consists of all real numbers of the form $a + b\sqrt{2}$ with a and b rational). If all of the conjugates of α are in the field $\mathbf{F}(\alpha)$, then $\mathbf{F}(\alpha)$ is called a **Galois** extension of $\mathbf{F}$.

Definition 3.1.6. The **derivative** of a polynomial is defined using the nX^{n-1} rule (not as a limit, since limits don't make sense in $\mathbf{F}$ unless there is a concept of distance or a topology in $\mathbf{F}$). A polynomial f of degree d may or may not have a root $r \in \mathbf{F}$, i.e., a value which gives 0 when substituted in place of X in the polynomial. If it does, then the degree-1 polynomial $X - r$ divides f; if $(X - r)^m$ is the highest power of $X - r$ which divides f, then we say that r is a root of **multiplicity m**. Because of unique factorization, the total number of roots of f in $\mathbf{F}$, counting multiplicity, cannot exceed d. If a polynomial $f \in \mathbf{F}[X]$ has a multiple root r, then r will be a root of the greatest common divisor of f and its derivative f'.

Definition 3.1.7. Given any polynomial $f(X) \in \mathbf{F}[X]$, there is an extension field $\mathbf{K}$ of $\mathbf{F}$ such that $f(X)$ splits into a product of linear factors (equivalently, has d roots in $\mathbf{K}$, counting multiplicity, where d is its degree) and such that $\mathbf{K}$ is the smallest extension field containing those roots. $\mathbf{K}$ is called the *splitting field* of f. The splitting field is unique

up to **isomorphism**, meaning that if we have any other field $\mathbf{K}'$ with the same properties, then there must be a 1-to-1 correspondence $\mathbf{K} \xrightarrow{\sim} \mathbf{K}'$ which preserves addition and multiplication. For example, $\mathbf{Q}(\sqrt{2})$ is the splitting field of $f(X) = X^2 - 2$, and to obtain the splitting field of $f(X) = X^3 - 2$ one must adjoin to $\mathbf{Q}$ both $\sqrt[3]{2}$ and $\sqrt{-3}$.

Definition 3.1.8. If adding the multiplicative identity 1 to itself in $\mathbf{F}$ never gives 0, then we say that $\mathbf{F}$ has **characteristic zero**; in that case $\mathbf{F}$ contains a copy of the field of rational numbers. Otherwise, there is a prime number p such that $1 + 1 + \cdots + 1$ (p times) equals 0, and p is called the *characteristic* of the field F. In that case F contains a copy of the field $\mathbf{Z}/p\mathbf{Z}$ (see Corollary 1 of Proposition 2.1.2), which is called its **prime field**.

Let Us Sum Up

- A vector space has a basis, and the number of elements in the basis is called its dimension.
- The polynomial ring F$[X]$ consists of all finite sums of powers of X with coefficients in F.
- The polynomial ring has unique factorization.
- An element α in some extension field $\mathbf{K}$ containing $\mathbf{F}$ is said to be algebraic over $\mathbf{F}$ if it satisfies a polynomial with coefficients in $\mathbf{F}$.
- The derivative of a polynomial is defined using the nX^{n-1} rule (not

as a limit, since limits don't make sense in $\mathbf{F}$ unless there is a concept of distance or a topology in $\mathbf{F}$).

- If $(X-r)^m$ is the highest power of $X-r$ which divides f, then we say that r is a root of multiplicity m.
- The splitting field is unique up to isomorphism.

Check you progress 3.1

1. Give any two irreducible polynomials of degree two over a field Z_3.
2. Find the splitting field of x^4+1 over the field of rational numbers Q.
3. Is $\pi+2$ algebraic over Q?

3.2 Finite Fields

Let $\mathbf{F}_q$ denote a field which has a finite number q of elements in it. Clearly a finite field cannot have characteristic zero; so let p be the characteristic of $\mathbf{F}_q$. Then $\mathbf{F}_q$ contains the prime field $\mathbf{F}_p = \mathbf{Z}/p\mathbf{Z}$, and so is a vector space – necessarily finite dimensional – over $\mathbf{F}_p$. Let f denote its dimension as an $\mathbf{F}_p$– vector space. Since choosing a basis enables us to set up a 1-to-1 correspondence between the elements of this f-dimensional vector space and the set of all f-tuples of elements in $\mathbf{F}_p$, it follows that there must be p^f elements in $\mathbf{F}_q$. That is, *q is a power of the characteristic p.*

We shall soon see that for every prime power $q=p^f$ there is a field of q elements, and it is unique (up to isomorphism).

But first we investigate the multiplicative **order** of elements in $\mathbf{F}_q^*$, the set of nonzero elements of our finite field. By the "order" of a nonzero element we mean the least positive power which is 1.

Existence of multiplicative generators of finite field.

There are $q-1$ nonzero elements, and, by the definition of a field, they form an *abelian group* with respect to multiplication. This means that the product of two nonzero elements is nonzero, the associative law and commutative law hold, there is an identity element 1, and any nonzero elements has an inverse. It is a general fact about finite groups that the order of any element must divide the number of elements in the group. For the sake of completeness, we give a proof of this in the case of our group $\mathbf{F}_q^*$.

Proposition 3.2.1. *The order of any $a \in \mathbf{F}_q^*$ divides $q-1$.*

First proof. Let d be the smallest power of a which equals 1. (Note that there is a finite power of a that is 1, since the powers of a in the finite set $\mathbf{F}_q^*$, cannot all be distinct, and as soon as $a^i = a^j$ for $j > i$ we have $a^{j-i} = 1$.) Let $\mathrm{S} = \{1, a, a^2, \cdots, a^{d-1}\}$ denote the set of all powers of a, and for any $b \in \mathbf{F}_q^*$ let bS denote the "coset" consisting of all elements of the form ba^j (for example, $1S = S$). It is easy to see that any two cosets are either identical or distinct (namely: if some b_1a^i in b_1S is also in b_2S, i.e., if it is of the form b_2a^j, then *any* element $b_1a^{i'}$ in b_1S is of the form to be in b_2S, because $b_1a^{i'} = b_1a^ia^{i'-i} = b_2a^{j+i'-i}$). And each coset

contains exactly d elements. Since the union of all the cosets exhausts $\mathbf{F}_q^*$, this means that $\mathbf{F}_q^*$; is a disjoint union of d-element sets; hence $d|(q-1)$.

Second proof. First we show that $a^{q-1} = 1$. To see this, write the product of all nonzero elements in $\mathbf{F}_q$. There are $q-1$ of them. If we multiply each of them by a, we get a rearrangement of the same elements (since any two distinct elements remain distinct after multiplication by a). Thus, the product is not affected. But we have multiplied this product by a^{q-1}. Hence $a^{q-1} = 1$. (Compare with the proof of Proposition 2.1.8.) Now let d be the order of a, i.e., the smallest positive power which gives 1. If d did not divide $q-1$, we could find a smaller positive number r, namely, the remainder when $q-1 = bd+r$ is divided by d, such that $a^r = a^{q-1-bd} = 1$. But this contradicts the minimality of d. This concludes the proof.

Definition 3.2.2. A **generator g** of a finite field $\mathbf{F}_q$ is an element of order $q-1$; equivalently, the powers of g run through all of the elements of $\mathbf{F}_q^*$.

The next proposition is one of the very basic facts about finite fields. It says that the nonzero elements of any finite field form a **cyclic group**, i.e., they are all powers of a single element.

Proposition 3.2.3. *Every finite field has a generator. If g is a generator of $\mathbf{F}_q^*$, then g^j is also a generator if and only if g.c.d.$(j, q-1) = 1$. In*

particular, there are a total of $\varphi(q-1)$ *different generators of* $\mathbf{F}_q^*$.

Proof. Suppose that $a \in \mathbf{F}_q^*$ has order d, i.e., $a^d = 1$ and no lower power of a gives 1. By Proposition 3.2.1, d divides $q-1$. Since a^d is the smallest power which equals 1, it follows that the elements $a, a^2, \cdots, a^d = 1$ are distinct. We claim that the elements of order d are precisely the $\varphi(d)$ values a^j for which $g.c.d.(j,d) = 1$. First, since the d distinct powers of a all satisfy the equation $x^d = 1$, these are all of the roots of the equation (see Definition 3.1.6). Any element of order d must thus be among the powers of a. However, not all powers of a have order d, since if $g.c.d.(j,d) = d' > 1$, then a^j has lower order: because d/d' and j/d' are integers, we can write $(a^j)^{(d/d')} = (a^d)^{j/d'} = 1$.
Conversely, we now show that a^j does have order d whenever $g.c.d.(j,d) = 1$. If j is prime to d, and if a^j had a smaller order d'', then $a^{d''}$ raised to either the j-th or the d-th power would give 1, and hence $a^{d''}$ raised to the power $g.c.d.(j,d) = 1$ would give 1 (this is proved in exactly the same way as Proposition 2.2.4). But this contradicts the fact that a is of order d and so $a^{d''} \neq 1$. Thus, a^j has order d if and only if $g.c.d.(j,d) = 1$.

This means that, if there is any element a of order d, then there are exactly $\varphi(d)$ elements of order d. So for every $d|(q-1)$ there are only two possibilities: no element has order d, or exactly $\varphi(d)$ elements have order d.

Now every element has some order $d|(q-1)$. And there are either 0 or $\varphi(d)$ elements of order d. But, by Proposition 2.1.21, $\sum_{d|(q-1)} \varphi(d) = q-1$,

which is the number of elements in $\mathbf{F}_q^*$. Thus, the only way that every element can have some order $d|(q-1)$ is if there are always $\varphi(d)$ (and never 0) elements of order d. In particular, there are $\varphi(q-1)$ elements of order $q-1$; and, as we saw in the previous paragraph, if g is any elements of order $q-1$, then the other elements of order $q-1$ are precisely the powers g^j for which $g.c.d.(j, q-1) = 1$. This completes the proof.

Corollary 3.2.4. *For every prime p, there exists an integer g such that the powers of g exhaust all nonzero residue classes modulo p.*

Example 3.2.5. We can get all residues mod 19 from 1 to 18 by taking powers of 2. Namely, the successive powers of 2 reduced mod 19 are: 2, 4, 8, 16, 13, 7, 14, 9, 18, 17, 15, 11, 3, 6, 12, 5, 10, 1. In many situations when working with finite fields, such as $\mathbf{F}_p$ for some prime p, it is useful to find a generator. What if a number $g \in \mathbf{F}_p^*$; is chosen at random? What is the probability that it will be a generator? In other words, what proportion of all of the nonzero elements consists of generators? According to Proposition 3.2.3, the proportion is $\varphi(p-1)/(p-1)$. But by our formula for $\varphi(n)$ following the corollary of Proposition 2.1.11, this fraction is equal to the $\prod(1 - \frac{1}{\updownarrow})$, where the product is over all primes $\updownarrow$ dividing $p-1$. Thus, the odds of getting a generator by a random guess depend heavily on the factorization of $p-1$.

Proposition 3.2.6. *There exists a sequence of primes p such that the probability that a random $g \in \mathbf{F}_p^*$ is a generator approaches zero.*

Proof. Let $\{n_j\}$ be any sequence of positive integers which is divisible

by more and more of the successive primes 2, 3, 5, 7,$\cdots$ as $j \longrightarrow \infty$. For example, we could take $n_j = j!$. Choose p_j to be any prime such that $p_j \equiv 1 \mod n_j$. How do we know that such a prime exists? That follows from **Dirichlet's theorem on primes in an arithmetic progression**, which states: **If n and k are relatively prime, then there are infinitely many primes which are $\equiv k \mod n$.** (In fact, more is true: the primes are "evenly distributed" among the different possible $k \mod n$, i.e., the proportion of primes $\equiv k \mod n$ is $1/\varphi(n)$; but we don't need that fact here.) Then the primes dividing $p_j - 1$ include all of the primes dividing n_j, and so $\frac{\varphi(p_j-1)}{p_j-1} \leq \Pi_{primes\, \updownarrow | n_j}(1 - \frac{1}{\updownarrow})$. But as $j \longrightarrow \infty$ this product approaches $\Pi_{all\ primes\, \updownarrow}(1 - \frac{1}{\updownarrow})$, which is zero. This proves the proposition.

Existence and uniqueness of finite fields with prime power number of elements. We prove both existence and uniqueness by showing that a finite field of $q = p^f$ elements is the splitting field of the polynomial $X^q - X$. The following proposition shows that for every prime power q there is one and (up to isomorphism) only one finite field with q elements.

Proposition 3.2.7. *If $\mathbf{F}_q$, is a field of $q = p^f$ elements, then every element satisfies the equation $X^q - X = 0$, and $\mathbf{F}_q$ is precisely the set of roots of that equation. Conversely, for every prime power $q = p^f$ the splitting field over $\mathbf{F}_p$ of the polynomial $X^q - X$ is a field of q elements*

Proof. First suppose that $\mathbf{F}_q$ is a finite field. Since the order of any nonzero element divides $q - 1$, it follows that any nonzero element satis-

fies the equation $X^{q-1} = 1$, and hence, if we multiply both sides by X, the equation $X^q = X$. Of course, the element 0 also satisfies the latter equation. Thus, all q elements of $\mathbf{F}_q$, are roots of the degree-q polynomial $X^q - X$. Since this polynomial cannot have more than q roots, its roots are precisely the elements of $\mathbf{F}_q$. This means that $\mathbf{F}_q$ is the splitting field of the polynomial $X^q - X$, that is, the smallest field extension of $\mathbf{F}_p$ which contains all of its roots.

Conversely, let $q = p^f$ be a prime power, and let $\mathbf{F}$ be the splitting field over $\mathbf{F}_p$ of the polynomial $X^q - X$. Note that $X^q - X$ has derivative $qX^{q-1} - 1 = -1$ (because the integer q is a multiple of p and so is zero in the field $\mathbf{F}_p$); hence, the polynomial $X^q - X$ has no common roots with its derivative (which has no roots at all), and therefore has no multiple roots. Thus, $\mathbf{F}$ must contain at least the q distinct roots of $X^q - X$. But we claim that the set of q roots is already a field. The key point is that a sum or product of two roots is again a root. Namely, if a and b satisfy the polynomial, we have $a^q = a$, $b^q = b$, and hence $(ab)^q = ab$, i.e., the product is also a root. To see that the sum $a + b$ also satisfies the polynomial $X^q - X = 0$, we note a fundamental fact about any field of characteristic p:

Lemma 3.2.8. *$(a + b)^p = a^p + b^p$ in any field of characteristic p.*

Proof. The lemma is proved by observing that all of the intermediate terms vanish in the binomial expansion $\sum_{j=0}^{p} \binom{p}{j} a^{p-j} b^j$, because $p!/(p-j)!j!$ is divisible by p for $0 < j < p$.

Repeated application of the lemma gives us: $a^p+b^p=(a+b)^p$, $a^{p^2}+b^{p^2}$ $=(a^p+b^p)^p=(a+b)^{p^2},\cdots,a^q+b^q=(a+b)^q$. Thus, if $a^q=a$ and $b^q=b$ it follows that $(a+b)^q=a+b$, and so $a+b$ is also a root of X^q-X. We conclude that the set of q roots is the smallest field containing the roots of X^q-X, i.e., the splitting field of this polynomial is a field of q elements. This completes the proof.

In the proof we showed that raising to the p-th power preserves addition and multiplication. We derive another important consequence of this in the next proposition.

Proposition 3.2.9. *Let $\mathbf{F}_p$, be the finite field of $q=p^f$ elements, and let σ be the map that sends every element to its p-th power: $\sigma(a)=a^p$. Then σ is an **automorphism** of the field $\mathbf{F}_q$ (a 1-to-1 map of the field to itself which preserves addition and multiplication). The elements of $\mathbf{F}_q$ which are kept fixed by σ are precisely the elements of the prime field $\mathbf{F}_p$. The f-th power (and no lower power) of the map σ is the identity map.*

Proof. A map that raises to a power always preserves multiplication. The fact that σ preserves addition comes from the proof of Lemma 3.2.8. Notice that for any j the j-th power of σ (the result of repeating σ j times) is the map $a\mapsto a^{p^j}$. Thus, the elements left fixed by σ^j are the roots of $X^{p^j}-X$. If $j=1$, these are precisely the p elements of the prime field (this is the special case $q=p$ of Proposition 3.2.7, namely, Fermat's Little Theorem). The elements left fixed by σ^f are the roots of X^q-X, i.e., all of $\mathbf{F}_q$. Since the f-th power of σ is the identity map, σ must be 1-to-1 (its inverse map is σ^{f-1}: $a\mapsto a^{p^{f-1}}$). No lower power of σ gives the

identity map, since for $j < f$ not all of the elements of $\mathbf{F}_q$ could be roots of the polynomial $X^{p^j} - X$. This completes the proof.

Proposition 3.2.10. *In the notation of Proposition 3.2.9, if α is any element of $\mathbf{F}_q$, then the conjugates of α over $\mathbf{F}_p$ (the elements of $\mathbf{F}_q$ which satisfy the same monic irreducible polynomial with coefficients in $\mathbf{F}_p$) are the elements $\sigma^j(\alpha) = \alpha^{p^j}$.*

Proof. Let d be the degree of $\mathbf{F}_p(\alpha)$ as an extension of $\mathbf{F}_p$. That is $\mathbf{F}_p(\alpha)$ is a copy of $\mathbf{F}_{p^d}$. Then α satisfies $X^{p^d} - X$ but does not satisfy $X^{p^j} - X$ for any $j < d$. Thus, one obtains d distinct elements by repeatedly applying σ to α. It now suffices to show that each of these elements satisfies the same monic irreducible polynomial $f(X)$ that α does, in which case they must be the d roots. To do this, it is enough to prove that, if α satisfies a polynomial $f(X) \in \mathbf{F}_p[X]$, then so does α^p. Let $f(X) = \sum a_j X^j$, where $a_j \in \mathbf{F}_p$. Then $0 = f(\alpha) = \sum a_j \alpha^j$. Raising both sides to the p-th power gives $0 = \sum (a_j \alpha^j)^p$ (where we use the fact that raising a sum $a + b$ to the p-th power gives $a^p + b^p$). But $a_j^p = a_j$, by Fermat's Little Theorem, and so we have: $0 = \sum a_j (\alpha^p)^j = f(\alpha^p)$, as desired. This completes the proof.

Explicit construction. So far our discussion of finite fields has been rather theoretical. Our only practical experience has been with the finite fields of the form $\mathbf{F}_p = \mathbf{Z}/p\mathbf{Z}$. We now discuss how to work with finite extensions of $\mathbf{F}_p$. At this point we should recall how in the case of the rational numbers $\mathbf{Q}$ we work with an extension such as $\mathbf{Q}(\sqrt{\mathbf{2}})$. Namely, we get this field by taking a root α of the equation $X^2 - 2$ and looking at expressions of the form $a + b\alpha$, which are added and multiplied in the

usual way, except that α^2 should always be replaced by 2. (In the case of $\mathbf{Q}(\sqrt[3]{2})$ we work with expressions of the form $a + b\alpha + c\alpha^2$, and when we multiply we always replace α^3 by 2.) We can take the same general approach with finite fields.

Example 3.2.11. To construct $\mathbf{F}_9$ we take any monic quadratic polynomial in $\mathbf{F}_3[X]$ which has no roots in $\mathbf{F}_3$. By trying all possible choices of coefficients and testing whether the elements $0, \pm 1 \in \mathbf{F}_3$ are roots, we find that there are three monic irreducible quadratics: X^2+1, $X^2 \pm X - 1$. If, for example, we take α to be a root of $X^2 + 1$ (let's call it i rather than α- after all, we are simply adjoining a square root of -1), then the elements of $\mathbf{F}_9$ are all combinations $a + bi$, where a and b are 0, 1, or -1. Doing arithmetic in $\mathbf{F}_9$ is thus a lot like doing arithmetic in the Gaussian integers (the complex numbers whose real and imaginary parts are integers), except that our arithmetic with the coefficients a and b occurs in the tiny field $\mathbf{F}_3$.

Notice that the element i that we adjoined is not a generator of $\mathbf{F}_9^*$, since it has order 4 rather than $q - 1 = 8$. If, however, we adjoin a root α of $X^2 - X - 1$, we can get all nonzero elements of $\mathbf{F}_9$ by taking the successive powers of α (remember that α^2 must always be replaced by $\alpha + 1$, since α satisfies $X^2 = X + 1$): $\alpha^1 = \alpha$, $\alpha^2 = \alpha + 1$, $\alpha^3 = -\alpha + 1$, $\alpha^4 = -1$, $\alpha^5 = -\alpha$, $\alpha^6 = -\alpha - 1$, $\alpha^7 = \alpha - 1$, $\alpha^8 = 1$. We sometimes say that the polynomial $X^2 - X - 1$ is *primitive*, meaning that any root of the irreducible polynomial is a generator of the group of nonzero elements of the field. There are $4 = \varphi(8)$ generators of $\mathbf{F}_9^*$, by Proposition 3.2.3: two

are the roots of $X^2 - X - 1$ and two are the roots of $X^2 + X - 1$. (The second root of $X^2 - X - 1$ is the conjugate of α, namely, $\sigma(\alpha) = \alpha^3 = -\alpha + 1$.) Of the remaining four nonzero elements, two are the roots of $X^2 + 1$ (namely $\pm i = \pm(\alpha + 1)$) and the other two are the two nonzero elements ± 1 of $\mathbf{F}_3$ (which are roots of the degree-1 monic irreducible polynomials $X - 1$ and $X + 1$.

In general, in any finite field $\mathbf{F}_q$, $q = p^f$, each element α satisfies a unique monic irreducible polynomial over $\mathbf{F}_p$, of some degree d. Then the field $\mathbf{F}_p(\alpha)$ obtained by adjoining this element to the prime field is an extension of degree d that is contained in $\mathbf{F}_q$. That is, it is a copy of the field $\mathbf{F}_{p^d}$. Since the big field $\mathbf{F}_{p^f}$ contains $\mathbf{F}_{p^d}$, and so is an $\mathbf{F}_{p^d}$-vector space of some dimension f', it follows that the number of elements in $\mathbf{F}_{p^f}$ must be $(p^d)^{f'}$, i.e., $f = df'$. Thus, $d|f$. Conversely, for any $d|f$ the finite field $\mathbf{F}_{p^d}$ is contained in $\mathbf{F}_q$, because any solution of $X^{p^d} = X$ is also a solution of $X^{p^f} = X$. (To see this, note that for any d', if you repeatedly replace X by X^{p^d} on the left in the equation $X^{p^d} = X$, you can obtain $X^{p^{dd'}} = 1$.) Thus, we have proved:

Proposition 3.2.12. *The subfields of* $\mathbf{F}_q$ *are the*$\mathbf{F}_{p^d}$ *for* d *dividing* f. *If an element of* $\mathbf{F}_{p^f}$ *is adjoined to* $\mathbf{F}_p$, *one obtains one of these fields.*

It is now easy to prove a formula that is useful in determining the number of irreducible polynomials of a given degree.

Proposition 3.2.13. *For any* $q = p^f$ *the polynomial* $X^q - X$ *factors in* $\mathbf{F}_p[X]$ *into the product of all monic irreducible polynomials of degrees* d

dividing f.

Proof. If we adjoin to $\mathbf{F}_p$ a root α of any monic irreducible polynomial of degree $d|f$, we obtain a copy of $\mathbf{F}_{p^d}$ which is contained in $\mathbf{F}_{p^f}$. Since α then satisfies $X^q - X = 0$, the monic irreducible must divide that polynomial. Conversely, let $f(X)$ be a monic irreducible polynomial which divides $X^q - X$. Then $f(X)$ must have its roots in $\mathbf{F}_p$ (since that's where all of the roots of $X^q - X$ are). Thus $f(X)$ must have degree dividing f, by Proposition 3.2.12, since adjoining a root gives a subfield of $\mathbf{F}_q$. Thus, the monic irreducible polynomials which divide $X^q - X$ are precisely all of the ones of degree dividing f. Since we saw that $X^q - X$ has no multiple factors, this means that $X^q - X$ is equal to the product of all such irreducible polynomials, as was to be proved.

Corollary 3.2.14. *If f is a prime number, then there are $(p^f - p)/f$ distinct monic irreducible polynomials of degree f in $\mathbf{F}_p[X]$.*

Proof. Notice that $(p^f - p)/f$ is an integer because of Fermat's Little Theorem for the prime f, which guarantees that $p^f \equiv p \mod f$. To prove the corollary, let n be the number of monic irreducible polynomials of degree f. According to the proposition, the degree-p^f polynomial $X^{p^f} - X$ is the product of n polynomials of degree f and the p degree-1 irreducible polynomials $X - a$ for $a \in \mathbf{F}_p$. Thus, equating degrees gives: $p^f = nf + p$, from which the desired equality follows.

More generally, suppose that f is not necessarily prime. Then, letting n_d denote the number of monic irreducible polynomials of degree d over $\mathbf{F}_p$, we have $n_f = (p^f - \sum dn_d)/f$, where the summation is over all $d < f$

which divide f.

We now extend the time estimates in unit 1 for arithmetic modulo p to general finite fields.

Proposition 3.2.15. *Let $\mathbf{F}_q$, where $q = p^f$, be a finite field, and let $F(X)$ be an irreducible polynomial of degree f over $\mathbf{F}_p$. Then two elements of $\mathbf{F}_q$ can be multiplied or divided in $O(log^3 q)$ bit operations. If k is a positive integer, then an element of $\mathbf{F}_q$ can be raised to the k-th power in $O(\log k\ log^3 q)$ bit operations.*

Proof. An element of $\mathbf{F}_q$ is a polynomial with coefficients in $\mathbf{F}_p = \mathbf{Z}/p\mathbf{Z}$ regarded modulo $F(X)$. To multiply two such elements, we multiply the polynomials - this requires $O(f^2)$ multiplications of integers modulo p (and some additions of integers modulo p, which take much less time) - and then divide the polynomial $F(X)$ into the product, taking the remainder polynomial as our answer. The polynomial division involves $O(f)$ divisions of integers modulo p and $O(f^2)$, multiplications of integers modulo p. Since a multiplication modulo p takes $O(log^2 p)$ bit operations, and a division (using the Euclidean algorithm, for example) takes $O(log^3 p)$ bit operations (see the corollary to Proposition 1.2.4), the total number of bit operations is: $O(f^2\ log^2 p + f\ 1og^3 p) = O((f\ logp)^3) = O(log^3 q)$. To prove the same result for division, it suffices to show that the reciprocal of an element can be found in time $O(log^3 q)$. Using the Euclidean algorithm for polynomials over the field $\mathbf{F}_p$, we must write 1 as a linear combination of our given element in $\mathbf{F}_q$ (i.e., a given polynomial of degree $< f$) and the fixed degree f polynomial $F(X)$. This involves $O(f)$ divisions of polyno-

mials of degree $< f$, and each polynomial division requires $O(f^2\ log^2 p + f\ log^3 p) = O(f^2\ log^3 p)$ bit operations. Thus, the total time required is $O(f^3\ log^3 p) = O(log^3 q)$. Finally, a k-th power can be computed by the repeated squaring method in the same way as modular exponentiation (see the end of first section in Unit 2). This takes $O(logk)$ multiplications (or squarings) of elements of $\mathbf{F}_q$, and hence $O(logk\ log^3 q)$ bit operations. This completes the proof.

We conclude this section with an example of computation with polynomials over finite fields. We illustrate by an example over the very smallest (and perhaps the most important) finite field, the 2-element field $\mathbf{F}_2 = \{0, 1\}$. A polynomial in $\mathbf{F}_2[X]$ is simply a sum of powers of X. In some ways, polynomials over $\mathbf{F}_p$ are like integers expanded to the base p, where the digits are analogous to the coefficients of the polynomial. For example, in its binary expansion an integer is written as a sum of powers of 2 (with coefficients 0 or 1), just as a polynomial over $\mathbf{F}_2$ is a sum of powers of X. But the comparison is often misleading. For example, the sum of any number of polynomials of degree d is a polynomial of degree (at most) d; whereas a sum of several d-bit integers will be an integer having more than d binary digits.

Example 3.2.16. Let $f(X) = X^4+X^3+X^2+1$, $g = X^3+1 \in \mathbf{F}_2[X]$. Find $g.c.d.(f, g)$ using the Euclidean algorithm for polynomials, and express the $g.c.d.$ in the form $u(X)f(X) + v(X)g(X)$.

Solution. Polynomial division gives us the sequence of equalities below which lead to the conclusion that $g.c.d.(f, g) = X + 1$, and the next sequence of equalities enables us, working backwards, to express $X + 1$ as a linear combination of f and g. (Note, by the way, that in a field of characteristic 2 adding is the same as subtracting, i.e., $a - b = a + b - 2b = a + b$.) We have:

$$\begin{aligned} f &= (X+1)g + (X^2 + X) \\ g &= (X+1)(X^2 + X) + (X+1) \\ X^2 + X &= X(X+1) \end{aligned}$$

and then

$$\begin{aligned} X + 1 &= g + (X+1)(X^2 + X) \\ &= g + (X+1)(f + (X+1)g) \\ &= (X+1)f + (X^2)g. \end{aligned}$$

Let Us Sum Up

- A finite field cannot have characteristic zero.
- For every prime power $q = p^f$, there is a field of q elements and it is unique upto isomorphism.
- The order of any $a \in \mathbf{F}_q^*$ divides $q - 1$.
- A generator g of a finite field $\mathbf{F}_q$ is an element of order $q - 1$.
- The nonzero elements of any finite field form a cyclic group.

- For every prime p, there exists an integer g such that the powers of g exhaust all nonzero residue classes modulo p.
- There exists a sequence of primes p such that the probability that a random $g \in \mathbf{F}_p^*$ is a generator approaches zero.
- If n and k are relatively prime, then there are infinitely many primes which are $\equiv k \mod n$.
- For every prime power q there is one and only one finite field with q elements.
- $(a+b)^p = a^p + b^p$ in any field of characteristic p.
- If f is a prime number, then there are $(p^f - p)/f$ distinct monic irreducible polynomials of degree f in $\mathbf{F}_p[X]$.

Check your progress 3.2

1. For p= 2, 3, 5, 7, 11, 13 and 17, find the smallest positive integer which generates $\mathbf{F}_p^*$, and determine how many of the integers 1, 2, 3, $\cdots$, $p-1$ are generators.
2. How many elements are in the smallest field extension of $\mathbf{F}_5$ which contains all of the roots of the polynomials X^2+X+1 and X^3+X+l?
3. For each degree $d \leq 6$, find the number of irreducible polynomials over $\mathbf{F}_2$ of degree d, and make a list of them.
4. For each degree $d \leq 6$, find the number of monic irreducible polynomials over $\mathbf{F}_3$ of degree d, and for $d \leq 3$ make a list of them.

5. Use the polynomial version of the Euclidean algorithm to find $g.c.d.(f, g)$ for $f, g \in \mathbf{F}_p[X]$ in each of the following examples. In each case express the $g.c.d.$ polynomial as a combination of f and g, i.e., in the form $d(X) = u(X)f(X) + v(X)g(X)$.

   (a) $f = X^3 + X + 1$, $g = X^2 + X + 1$, $p = 2$;

   (b) $f = X^6+X^5+X^4+X^3+X^2+X+1$, $g = X^4+X^2+X+1$,$p = 2$;

   (c) $f = X^3 - X + 1$, $g = X^2 + 1$, $p = 3$;

   (d) $f = X^5 + X^4 + X^3 - X^2 - X + 1$,$g = X^3 + X^2 + X + 1$, $p = 3$;

   (e) $f = X^5+88X^4+73X^3+83X^2+51X+67$, $g = X^3+97X^2+40X+38$, $p = 101$.

3.3 Quadratic residues and reciprocity

Roots of unity. In many situations it is useful to have solutions of the equation $x^n = 1$. Suppose we are working in a finite field $\mathbf{F}_q$. We now answer the questions: How many n-th roots of unity are there in $\mathbf{F}_q$?

Proposition 3.3.1. *Let g be a generator of ${\mathbf{F}^*}_q$. Then g^j is an n-th root of unity if and only if $nj \equiv 0 \mod q-1$. The number of n-th roots of unity is $g.c.d.(n, q-1)$. In particular, $\mathbf{F}_q$ has a **primitive** n-th root of unity (i.e., an element ξ such that the powers of ξ run through n n-th roots of unity) if and only if $n|q-1$. If ξ is a primitive n-th root of unity in $\mathbf{F}_q$, then ξ^j is also a primitive n-th root if and only if $g.c.d.(j, n) = 1$.*

Proof. Any element of ${\mathbf{F}^*}_q$ can be written as a power g^j of the generator

g. A power of g is 1 if and only if the power is divisible by $q-1$. Thus, an element g^j is an n-th root of unity if and only if $nj \equiv 0 \mod q-1$. Next, let $d = g.c.d.(n, q-1)$. According to Corollary 2 of Proposition 2.1.2, the equation $nj \equiv 0 \mod q-1$ (with j the unknown) is equivalent to the equation $\frac{n}{d}j \equiv 0 \mod \frac{q-1}{d}$. Since n/d is prime to $(q-1)/d$, the latter congruence is equivalent to requiring j to be a multiple of $(q-1)/d$. In other words, the d distinct powers of $g^{(q-1)/d}$ are precisely the n-th roots of unity. There are n such roots if and only if $d = n$, i.e., $n|q-1$. Finally, if n does divide $q-1$, let $\xi = g^{(q-1)/n}$. Then ξ^j equals 1 if and only if $n|j$. The k-th power of ξ^j equals 1 if and only if $kj \equiv 0 \mod n$. It is easy to see that ξ^j has order n (i.e., this equation does not hold for any positive $k < n$) if and only if j is prime to n. Thus, there are $\varphi(n)$ different primitive n-th roots of unity if $n|q-1$. This completes the proof.

Corollary 3.3.2. *If $g.c.d.(n, q-1) = 1$, then 1 is the only n-th root of unity.*

Proof. By the above Proposition, the number of n-th roots of unity is $g.c.d.(n, q-1)$, which is 1. Thus, 1 is the only n-th root of unity.

Corollary 3.3.3. *The element $-1 \in \mathbf{F}_q$ has a square root in $\mathbf{F}_q$ if and only if $q \equiv 1 \mod 4$.*

A square root of -1 is the same thing as a primitive 4-th root of 1, and our field has a primitive 4-th root if and only if $4|q-1$.

Remark 3.3.4. Corollary 3.2.3 says that if $q \equiv 3 \mod 4$, we can always get the quadratic extension $\mathbf{F}_{q^2}$ by adjoining a root of $X^2 + 1$, i.e., by considering "Gaussian integer" type expressions $a + bi$. We did this for $q = 3$ in the last section.

Let us suppose, for example, that p is a prime which is $\equiv 3 \mod 4$. There is a nice way to think of the field $\mathbf{F}_{p^2}$ which generalizes to other situations. Let R denote the Gaussian integer ring. Sometimes we write $R = \mathbf{Z} + \mathbf{Z}i$, meaning the set of all integer combinations of 1 and i. If m is any Gaussian integer, and $\alpha = a + bi$ and $\beta = c + di$ are two Gaussian integers, we write $\alpha \equiv \beta \mod m$ if $\alpha - \beta$ is divisible by m, i.e., if the quotient is a Gaussian integer. We can then look at the set R/mR of residue classes modulo m; just as in the case of ordinary integers, residue classes can be added or multiplied, and the residue class of the result does not depend on which representatives were chosen for the residue class factors. Now if $m = p + 0i$ is a prime number which is $\equiv 3 \mod 4$, it is not hard to show that R/pR is the field $\mathbf{F}_{p^2}$.

Quadratic residues. Suppose that p is an odd prime, i.e., $p > 2$. We are interested in knowing which of the nonzero elements $\{1, 2, \cdots, p-1\}$ of $\mathbf{F}_p$ are squares. If some $a \in \mathbf{F}^*_p$ is a square, say $b^2 = a$, then a has precisely two square roots $\pm b$ (since the equation $X^2 - a = 0$ has at most two solutions in a field). Thus, the squares in $\mathbf{F}^*_p$ can all be found by computing $b^2 \mod p$ for $b = 1, 2, 3, \cdots, (p-1)/2$ (since the remaining integers up to $p - 1$ are all $\equiv -b$ for one of these b), and precisely half of the elements in $\mathbf{F}^*_p$ are squares. For example, the squares in $\mathbf{F}_{11}$ are 1^2

$= 1$, $2^2 = 4$, $3^2 = 9$, $4^2 = 5$, and $5^2 = 3$. The squares in $\mathbf{F}_p$ are called *quadratic residues* modulo p. The remaining nonzero elements are called *nonresidues*. For $p = 11$ the nonresidues are 2, 6, 7, 8, 10. There are $(p-1)/2$ residues and $(p-1)/2$ nonresidues.

If g is a generator of $\mathbf{F}_p$, then any element can be written in the form g^j. Thus, the square of any element is of the form g^j with j even. Conversely, any element of the form g^j with j even is the square of some element, namely $\pm g^{j/2}$.

Definition 3.3.5. (The Legendre symbol).

Let a be an integer and $p > 2$ a prime. We define the *Legendre symbol* $\left(\frac{a}{p}\right)$ to equal 0, 1 or -1, as follows:

$$\left(\frac{a}{p}\right) = \begin{cases} 0, & if\ p|a; \\ 1, & if\ a \text{ is a quadratic residue } \mod p; \\ -1 & if\ a \text{ is a nonresidue } \mod p. \end{cases}$$

Thus, the Legendre symbol is simply a way of identifying whether or not an integer is a quadratic residue modulo p.

Proposition 3.3.6. $\left(\frac{a}{p}\right) \equiv a^{(p-1)/2} \mod p$

Proof. If a is divisible by p, then both sides are $\equiv 0 \mod p$. Suppose $p \nmid a$. By Fermat's Little Theorem, in $\mathbf{F}_p$ the square of $a^{(p-1)/2}$ is 1, so $a^{(p-1)/2}$ itself is ± 1. Let g be a generator of $\mathbf{F}^*{}_p$, and let $a = g^j$. As we saw, a is a residue if and only if j is even. And $a^{(p-1)/2} = g^{j(p-1)/2}$ is 1 if and only if $j(p-1)/2$ is divisible by $p-1$, i.e., if and only if j is even.

Thus, both sides of the congruence in the proposition are ± 1 in $\mathbf{F}_p$, and each side is $+1$ if and only if j is even. This completes the proof.

Proposition 3.3.7. *The Legendre symbol satisfies the following properties:*

(a) $\left(\frac{a}{p}\right)$ *depends only on the residue of* a *modulo* p*;*

(b) $\left(\frac{ab}{p}\right) = \left(\frac{a}{p}\right)\left(\frac{b}{p}\right)$*;*

(c) for b *prime to* p*,* $\left(\frac{ab^2}{p}\right) = \left(\frac{a}{p}\right)$*;*

(d) $\left(\frac{1}{p}\right) = 1$ *and* $\left(\frac{-1}{p}\right) = (-1)^{(p-1)/2}$.

Proof. Part (a) is obvious from the definition. Part (b) follows from Proposition 3.3.6, because the right side is congruent modulo p to $a^{(p-1)/2} \cdot b^{(p-1)/2} = (ab)^{(p-1)/2}$, as is the left side. Part (c) follows immediately from part (b). The first equality in part (d) is obvious, because $1^2 = 1$, and the second equality comes from Corollary 3.3.3 (or by taking $a = -1$ in Proposition 3.3.6). This completes the proof.

Remark 3.3.8. Part (b) of Proposition 3.3.6 shows that one can determine if a number a is a quadratic residue modulo p, i.e., one can evaluate $\left(\frac{a}{p}\right)$, if one factors a and knows the Legendre symbol for the factors. The first step in doing this is to write a as a power of 2 times an odd number. We then want to know how to evaluate $\left(\frac{2}{p}\right)$.

Proposition 3.3.9.

$$\left(\tfrac{2}{p}\right) = (-1)^{(p^2-1)/8} = \begin{cases} 1 & if\ p \equiv \pm \mod 8; \\ -1 & if\ p \equiv \pm 3 \mod 8. \end{cases}$$

Proof. Let $f(n) = (-1)^{(n^2-1)/8}$ for n odd, $f(n) = 0$ for n even. We want to show that $\left(\frac{2}{p}\right) = f(p)$. Of the various ways of proving this, we shall use an efficient method based on what we already know about finite fields. Since $p^2 \equiv 1 \mod 8$ for any odd prime p, we know that the field $\mathbf{F}_{p^2}$ contains a primitive 8-th root of unity. Let $\xi \in \mathbf{F}_{p^2}$ denote a primitive 8-th root of 1. Note that $\xi^4 = -1$. Define $G = \sum_{j=0}^{7} f(j)\xi^j$. ($G$ is an example of what is called a *Gauss sum*.) Then $G = \xi - \xi^3 - \xi^5 + \xi^7 = 2(\xi - \xi^3)$ (because $\xi^5 = \xi^4\xi = -\xi$ and $\xi^7 = -\xi^3$), and $G^2 = 4(\xi^2 - 2\xi^4 + \xi^6) = 8$. Thus, in $\mathbf{F}_{p^2}$ we have

$$G^p = (G^2)^{(p-1)/2}G = 8^{(p-1)/2}G = \left(\tfrac{8}{p}\right)G = \left(\tfrac{2}{p}\right)G,$$

by Proposition 3.3.6 and Proposition 3.3.7(c). On the other hand, using the definition of G, the fact that $(a+b)^p = a^p + b^p$ in $\mathbf{F}_{p^2}$, and the obvious observation that $f(j)^p = f(j)$, we compute: $G^p = \sum_{j=0}^{7} f(j)\xi^{pj}$. Notice that $f(j) = f(p)f(pj)$, as we easily check. Then, making the change of variables $j' = pj$ (i.e., modulo 8 we have j' running through 0, $\cdots$,7 when j does), we obtain:

$$G^p = \sum_{j=0}^{7} f(p)f(pj)\xi^{pj} = f(p)\sum_{j'=0}^{7} f(j')\xi^{j'} = f(p)G.$$

Comparing the two equalities for G^p gives the desired result. (Notice that we can divide by G, since it is not 0 in $\mathbf{F}_{p^2}$, as is clear from the fact that

its square is 8.)

Remark 3.3.10. Next, we must deal with the odd prime factors of a. Let q stand for such an odd prime factor. i.e., q stand for an odd prime distinct from p, not for a power of p as in the last section.

Since a can be assumed to be smaller than p (by part (a) of Proposition 3.3.7), the prime factors q will be smaller than p. The next proposition - the fundamental Law of Quadratic Reciprocity - tells us how to relate $\left(\frac{q}{p}\right)$ to $\left(\frac{p}{q}\right)$. The latter Legendre symbol will be easier to evaluate, since we can immediately replace p by its least positive residue modulo q, thereby reducing ourselves to a Legendre symbol involving smaller numbers. The quadratic reciprocity law states that $\left(\frac{q}{p}\right)$ and $\left(\frac{p}{q}\right)$ are the same unless p and q are both $\equiv 3 \mod 4$, in which case they are the negatives of one another. This can be expressed as a formula using the fact that $(p-1)(q-1)/4$ is even unless both primes are $\equiv 3 \mod 4$, in which case it is odd.

Proposition 3.3.11. ***(Law of Quadratic Reciprocity).*** *Let p and q be two odd primes. Then*

$$\left(\frac{q}{p}\right) = (-1)^{(p-1)(q-1)/4}\left(\frac{p}{q}\right) = \begin{cases} -\left(\frac{p}{q}\right) & if\ p \equiv q \equiv 3 \mod 4; \\ \left(\frac{p}{q}\right) & otherwise \end{cases}$$

Proof. We shall give a particularly short proof along the lines of the proof of the last proposition, using finite fields. Let f be any power of p such

that $p^f \equiv 1 \mod q$. For example, we can always take $f = q-1$. Then, as we saw at the beginning of the section (Proposition 3.3.1), the field $\mathbf{F}_{p^f}$ contains a primitive q-th root of unity, which we denote ξ. (Remember that q here denotes another prime besides p; it does not denote p^f.) We define the "Gauss sum" G by the formula $G = \sum_{j=0}^{q-1} \left(\frac{j}{q}\right)\xi^j$. In the next paragraph we shall prove that $G^2 = (-1)^{(q-1)/2}q$. Before proving that lemma, we show how to use it to prove our proposition. The proof is very similar to the proof of Proposition 3.3.9. We first obtain (using the lemma to be proved below):

$$\begin{aligned} G^p &= (G^2)^{(p-1)/2}G = \left((-1)^{(q-1)/2}q\right)^{(p-1)/2}G \\ &= (-1)^{(p-1)(q-1)/4}q^{(p-1)/2}G = (-1)^{(p-1)(q-1)/4}\left(\frac{q}{p}\right)G \end{aligned}$$

by Proposition 3.3.6 with a replaced by q (recall that we're working in a field of characteristic p, namely $\mathbf{F}_{p^f}$, and so congruence modulo p becomes equality). On the other hand, using the definition of G, the fact that $(a+b)^p = a^p + b^p$ in $\mathbf{F}_{p^f}$, and the obvious observation that $\left(\frac{j}{q}\right)^p = \left(\frac{j}{q}\right)$, we compute:

$$G^p = \sum_{j=0}^{q-1} \left(\frac{j}{q}\right)\xi^{pj} = \sum_{j=0}^{q-1} \left(\frac{p}{q}\right)\left(\frac{pj}{q}\right)\xi^{pj},$$

by parts (b) and (c) of Proposition 3.3.7. Pulling $\left(\frac{p}{q}\right)$ outside the summation and making the change of variables $j' = pj$ in the summation, we finally obtain: $G^p = \left(\frac{p}{q}\right)G$. Equating our two expressions for G^p and dividing by G (which is possible, since $G^2 = \pm q$ and so is not zero in

$\mathbf{F}_{p^f}$), we obtain the quadratic reciprocity law. Thus, it remains to prove the following lemma.

Lemma 3.3.12. $G^2 = (-1)^{(q-1)/2}q$

Proof. Using the definition of G, where in one copy of G we replace the variable of summation j by $-k$ (and note that the summation can start at 1 rather than 0, since $\left(\frac{0}{q}\right) = 0$), we have:

$$\begin{aligned} G^2 &= \sum_{j,k=1}^{q-1} \left(\frac{j}{q}\right)\xi^j\left(\frac{-k}{q}\right)\xi^{-k} = \left(\frac{-1}{q}\right)\sum_{j=1}^{q-1}\sum_{k=1}^{q-1}\left(\frac{jk}{q}\right)\xi^{j-k} \\ &= (-1)^{(q-1)/2}\sum_{j=1}^{q-1}\sum_{k=1}^{q-1}\left(\frac{j^2k}{q}\right)\xi^{j(1-k)} \end{aligned}$$

where we have used Part (d) of Proposition 3.3.7 to replace $\left(\frac{-1}{q}\right)$ by $(-1)^{(q-1)/2}$, and for each value of j we have made a change of variable in the inner summation $k \longleftrightarrow kj$ (i.e., for each fixed j, kj runs through the residues modulo q as k does, and the summands depend only on the residue modulo q). We next use part (c) of Proposition 3.3.7, interchange the order of summation, and pull the $\left(\frac{k}{q}\right)$ outside the inner sum over j. The double sum then becomes $\sum_k \left(\frac{k}{q}\right)\sum_j \xi^{j(1-k)}$. Here both sums go from 1 to $q-1$, but if we want we can insert the terms with $j = 0$, since that simply adds to the double sum $\sum_k \left(\frac{k}{q}\right)$, which is zero (because there are equally many residues and nonresidues modulo q). Thus, the double sum can be written $\sum_{k=1}^{q-1}\left(\frac{k}{q}\right)\sum_{j=0}^{q-1}\xi^{j(1-k)}$. But for each k other than 1, the inner sum vanishes. This is because the sum of the distinct powers of a nontrivial ($\neq 1$) root of unity ξ' is zero (the simplest way to see this is to note that multiplying the sum by ξ' just rearranges it, and so the sum

multiplied by $\xi' - 1$ is zero). So we are left with the contribution when $k = 1$, and we finally obtain:

$$G^2 = (-1)^{(q-1)/2}\left(\tfrac{1}{q}\right)\Sigma_{j=0}^{q-1}\xi^0 = (-1)^{(q-1)/2}q$$

This completes the proof of the lemma, and hence also the proof of the Law of Quadratic Reciprocity.

Example 3.3.13. Determine whether 7411 is a residue modulo the prime 9283.

Solution. Since 7411 and 9283 are both primes which are $\equiv 3 \mod 4$, we have $\left(\frac{7411}{9283}\right) = -\left(\frac{9283}{7411}\right) = -\left(\frac{1872}{7411}\right)$ by part (a) of Proposition 3.3.7. Since $1872 = 2^4 \cdot 3^2 \cdot 13$, by part (c) of Proposition 3.3.7 we find that the desired Legendre symbol is $-\left(\frac{13}{7411}\right)$. But we can now apply quadratic reciprocity again: since $13 \equiv 1 \mod 4$ we find that $-\left(\frac{13}{7411}\right) = -\left(\frac{7411}{13}\right) = -\left(\frac{1}{13}\right) = -1$. In other words, 7411 is a quadratic nonresidue.

Remark 3.3.14. One difficulty with this method of evaluating Legendre symbols is that at each stage we must factor the number on top in order to apply Proposition 3.3.11. If our numbers are astronomically large, this will be very time consuming. Fortunately, it is possible to avoid any need for factoring (except taking out powers of 2, which is very easy), once we prove a generalization of the quadratic reciprocity law that applies to all positive odd integers, not necessarily prime. But we first need a definition which generalizes the definition of the Legendre symbol.

Definition 3.3.15. (The Jacobi symbol). Let a be an integer, and let n be any positive odd number. Let $n = p_1^{\alpha_1} \cdots p_r^{\alpha_r}$ be the prime factor-

ization of n. Then we define the *Jacobi symbol* $\left(\frac{a}{n}\right)$ as the product of the Legendre symbols for the prime factors of n:

$$\left(\frac{a}{n}\right) = \left(\frac{a}{p_1}\right)^{\alpha_1} \cdots \left(\frac{a}{p_r}\right)^{\alpha_r}.$$

A word of warning is in order here. If $\left(\frac{a}{n}\right) = 1$ for n composite, it is not necessarily true that a is a square modulo n. For example, $\left(\frac{2}{15}\right) = \left(\frac{2}{3}\right)\left(\frac{2}{5}\right)$ $= (-1)(-1) = 1$, but there is no integer x such that $x^2 \equiv 2 \mod 15$.

We now generalize Proposition 3.3.11 to the Jacobi symbol.

Proposition 3.3.16. *For any positive odd n we have $\left(\frac{2}{n}\right) = (-1)^{(n^2-1)/8}$.*

Proof. Let $f(n)$ denote the function on the right side of the equality, as in the proof of Proposition 3.3.11. It is easy to see that $f(n_1 n_2) = f(n_1)f(n_2)$ for any two odd numbers n_1 and n_2. (Just consider the different possibilities for n_1 and n_2 modulo 8.) This means that the right side of the equality in the proposition equals $f(p_1)^{\alpha_1} \cdots f(p_r)^{\alpha_r} = \left(\frac{2}{p_1}\right)^{\alpha_1} \cdots \left(\frac{2}{p_r}\right)^{\alpha_r}$ by Proposition 3.3.11. But this is $\left(\frac{2}{n}\right)$, by definition.

Proposition 3.3.17. *For any two positive odd integers m and n we have $\left(\frac{m}{n}\right) = (-1)^{(m-1)(n-1)/4}\left(\frac{n}{m}\right)$.*

Proof. First note that if m and n have a common factor, then it follows from the definition of the Legendre and Jacobi symbols that both sides are zero. So we can suppose that $g.c.d.(m, n) = 1$. Next, we write m and n as products of primes: $m = p_1 p_2 \cdots p_r$ and $n = q_1 q_2 \cdots q_s$. (The p's and q's include repetitions if m or n has a square factor.) In converting from $\left(\frac{m}{n}\right) = \prod_{i,j} \left(\frac{p_i}{q_j}\right)$ to $\left(\frac{n}{m}\right) = \prod_{i,j} \left(\frac{q_j}{p_i}\right)$ we must apply the quadratic reciprocity

law for the Legendre symbol rs times. The number of (-1)'s we get is the number of times both p_i and q_j are $\equiv 3 \mod 4$, i.e., it is the product of the number of primes $\equiv 3 \mod 4$ in the factorization of m and in the factorization of n. Thus, $\left(\frac{m}{n}\right) = \left(\frac{n}{m}\right)$ unless there are an odd number of primes $\equiv 3 \mod 4$ in both factorizations, in which case $\left(\frac{m}{n}\right) = -\left(\frac{n}{m}\right)$. But a product of odd primes, such as m or n, is $\equiv 3 \mod 4$ if and only if it contains an odd number of primes which are $\equiv 3 \mod 4$. We conclude that $\left(\frac{m}{n}\right) = \left(\frac{n}{m}\right)$ unless both m and n are $\equiv 3 \mod 4$, as was to be proved. This gives us the reciprocity law for the Jacobi symbol.

Example 3.3.18. We return to Example 3.3.13, and show how to evaluate the Legendre symbol without factoring 1872, except to take out the power of 2. By the reciprocity law for the Jacobi symbol we have

$$-\left(\frac{1872}{7411}\right) = -\left(\frac{16}{7411}\right)\left(\frac{117}{7411}\right) = -\left(\frac{7411}{117}\right) = -\left(\frac{40}{117}\right),$$

and this is equal to $-\left(\frac{2}{117}\right)\left(\frac{5}{117}\right) = \left(\frac{5}{117}\right) = \left(\frac{117}{5}\right) = \left(\frac{2}{5}\right) = -1$.

Square roots modulo p. Using quadratic reciprocity, one can quickly determine whether or not an integer a is a quadratic residue modulo p. However, if it is a residue, that does not tell us how to find a solution to the congruence $x^2 \equiv a \mod p$, it tells us only that a solution exists. We conclude this section by giving an algorithm for finding a square root of a residue a once we know any nonresidue n.

Let p be an odd prime, and suppose that we somehow know a quadratic nonresidue n. Let a be an integer such that $\left(\frac{a}{p}\right) = 1$. We want to find

an integer x such that $x^2 \equiv a \mod p$. Here is how we proceed. First write $p-1$ in the form $2^\alpha \cdot s$, where s is odd. Then compute n^s modulo p, and call that b. Next compute $a^{(s+1)/2}$ modulo p, and call that r. Our first claim is that r comes reasonably close to being a square root of a. More precisely, if we take the ratio of r^2 to a, we claim that we get a $2^{\alpha-1}$-th root of unity modulo p. Namely, we compute (for brevity, we shall use equality to mean congruence modulo p, and we use a^{-1} to mean the inverse of a modulo p):

$$(a^{-1}r^2)^{2^{\alpha-1}} = a^{s2^{\alpha-1}} = a^{(p-1)/2} = \left(\frac{a}{p}\right) = 1.$$

We must then modify r by a suitable 2^α-th root of unity to get an x such that x^2/a is 1. To do this, we claim that b is a primitive 2^α-th root of unity, which means that all 2^α-th roots of unity are powers of b. To see this, first we note that b is a 2^α-th root of 1, because $b^{2^\alpha} = n^{2^\alpha s} = n^{p-1}$ $= 1$. If b weren't primitive, there would be a lower power (a divisor of 2^α) of b that gives 1. But then b would be an even power of a primitive 2^α-th root of unity, and so would be a square in $\mathbf{F}_p^*$. This is impossible, because $\left(\frac{b}{p}\right) = \left(\frac{n}{p}\right)^s = -1$ (since s is odd and n is a nonresidue). Thus, b is a primitive 2^α-th root of unity. So it remains to find a suitable power b^j, $0 \le j < 2^\alpha$, such that $x = b^j r$ gives the desired square root of a. To do that, we write j in binary as $j = j_0 + 2j_1 + 4j_2 + \cdots + 2^{\alpha-2}j_{\alpha-2}$, and show how one successively determines whether $j_0, j_1, \cdots$ is 0 or 1. (Note that we may suppose that $j < 2^{\alpha-1}$, since $b^{2^{\alpha-1}} = -1$, and so j can be modified by $2^{\alpha-1}$ to give another j for which $b^j r$ is the other square root

of a.) Here is the inductive procedure for determining the binary digits of j:

1. Raise (r^2/a) to the $2^{\alpha-2}$-th power. We proved that the square of this is 1. Hence, you get either ± 1. If you get 1, take $j_0 = 0$; if you get -1, take $j_0 = 1$. Notice that j_0 has been chosen so that $((b^{j_0}r)^2/a)$ is a $2^{\alpha-2}$-th root of unity.

2. Suppose you've found $j_0, \cdots, j_{k-1}$ such that $(b^{j_0+2j_1+\cdots+2^{k-1}j_{k-1}}r)^2/a$ is a $2^{\alpha-k-1}$ -th root of unity, and you want to find j_k. Raise this number to half the power that gives 1, and choose j_k according to whether you get $+1$ or -1:

$$\text{if } \left(\frac{(b^{j_0+2j_1+\cdots+2^{k-1}j_{k-1}}r)^2}{a}\right)^{2^{\alpha-k-2}} = \begin{cases} 1 \\ \\ -1 \end{cases},$$

then take $j_k = \begin{cases} 0 \\ \\ 1 \end{cases}$ respectively.

We easily check that with this choice of j_k the "corrected" value comes closer to being a square root of a, i.e., we find that $(b^{j_0+2j_1+\cdots+2^{k-1}j_k}r)^2/a$ is a $2^{\alpha-k-2}$-th root of unity.

When we get to $k = \alpha - 2$ and find $j_{\alpha-2}$, we then have

$$(b^{j_0+2j_1+\cdots+2^{\alpha-2}j_{\alpha-2}}r)^2/a = 1,$$

i.e., $b^j r$ is a square root of a, as desired.

Example 3.3.19. Use the above algorithm to find a square root of $a = 186$ modulo $p = 401$.

Solution. The first nonresidue is $n = 3$. We have $p - 1 = 2^4 \cdot 25$, and so $b = 3^{25} = 268$ and $r = a^{13} = 103$ (where we use equality to denote congruence modulo p). After first computing $a^{-1} = 235$, we note that $r^2/a = 98$, which must be an 8-th root of 1. We compute that $98^4 = -1$, and so $j_0 = 1$. Next, we compute $(br)^2/a = -1$. Since the 2-nd power of this is 1, we have $j_1 = 0$, and then $j_2 = 1$. Thus, $j = 5$ and the desired square root is $b^5 r = 304$.

Remark 3.3.20. The easiest case of this algorithm occurs when p is a prime which is $\equiv 3 \mod 4$. Then $\alpha = 1$, $s = (p-1)/2$, so $(s+1)/2 = (p+1)/4$, and we see that $x = r = a^{(p+1)/4}$ is already the desired square root.

Remark 3.3.21. We now discuss the time estimate for this algorithm. We suppose that we start already knowing the information that n is a nonresidue. The steps in finding s, b and $r = a^{(s+1)/2}$ (working modulo p, of course) take at most $O(log^3 p)$ bit operations (see Proposition 2.1.19). Then in finding j the most time-consuming part of the k-th induction step is raising a number to the $2^{\alpha-k-2}$-th power, and this means $\alpha - k - 2$ squarings $\mod p$ of integers less than p. Since $\alpha - k - 2 < \alpha$. we have the estimate $O(\alpha\ log^2 p)$ for each step. Thus, since there are $\alpha - 1$ steps, the final estimate is $O(log^3 p + \alpha^2 log^2 p) = O(log^2 p(logp + \alpha^2))$. At worst (if almost all of $p - 1$ is a power of 2), this is $O(log^4 p)$, since $\alpha < log_2 p =$ $O(log\ p)$ Thus, given a nonresidue modulo p, we can extract square roots

mod p in polynomial time (bounded by the fourth power of the number of bits in p).

Remark 3.3.22. Strictly speaking, it is not known (unless one assumes the validity of the so-called "Riemann Hypothesis") whether there is an algorithm for finding a nonresidue modulo p in polynomial time. However, given any $\epsilon > 0$ there is a polynomial time algorithm that finds a nonresidue with probability greater than 1 - ϵ. Namely, a randomly chosen number n, $0 < n < p$, has a 50% chance of being a nonresidue, and this can be checked in polynomial time. If we do this for more than $log_2(l/\epsilon)$ different randomly chosen n, then with probability $> 1 - \epsilon$ at least one of them will be a nonresidue.

Let Us Sum Up

- $\mathbf{F}_q$ has a primitive n-th root of unity if and only if $n|q-1$. If ξ is a primitive n-th root of unity in $\mathbf{F}_q$, then ξ^j is also a primitive n-th root if and only if $g.c.d.(j,n) = 1$.
- If $g.c.d.(n, q-1) = 1$, then 1 is the only n-th root of unity.
- The element $-1 \in \mathbf{F}_q$ has a square root in $\mathbf{F}_q$ if and only if $q \equiv 1 \mod 4$.
- The Legendre symbol is simply a way of identifying whether or not an integer is a quadratic residue modulo p.
- $\left(\frac{a}{p}\right) \equiv a^{(p-1)/2} \mod p$.

- (a) $\left(\frac{a}{p}\right)$ depends only on the residue of a modulo p;

  (b) $\left(\frac{ab}{p}\right) = \left(\frac{a}{p}\right)\left(\frac{b}{p}\right)$;

  (c) for b prime to p, $\left(\frac{ab^2}{p}\right) = \left(\frac{a}{p}\right)$;

  (d) $\left(\frac{1}{p}\right) = 1$ and $\left(\frac{-1}{p}\right) = (-1)^{(p-1)/2}$.

- $\left(\frac{2}{p}\right) = (-1)^{(p^2-1)/8} = \begin{cases} 1 & if\ p \equiv \pm \mod 8; \\ -1 & if\ p \equiv \pm 3 \mod 8. \end{cases}$

- Let p and q be two odd primes. Then

$$\left(\frac{q}{p}\right) = (-1)^{(p-1)(q-1)/4}\left(\frac{p}{q}\right) = \begin{cases} -\left(\frac{p}{q}\right) & if\ p \equiv q \equiv 3 \mod 4; \\ \left(\frac{p}{q}\right) & otherwise \end{cases}$$

- $G^2 = (-1)^{(q-1)/2}q$.

- $\left(\frac{a}{n}\right) = 1$, for n composite.

- For any positive odd n we have $\left(\frac{2}{n}\right) = (-1)^{(n^2-1)/8}$.

- For any two positive odd integers m and n we have $\left(\frac{m}{n}\right) = (-1)^{(m-1)(n-1)/4}\left(\frac{n}{m}\right)$.

Check your progress 3.3.

1. Make a table showing all quadratic residues and nonresidues modulo p for $p = 3, 5, 7, 13, 17, 19$.

2. How many 84-th roots of 1 are there in the field of 11^3 elements?

3. Prove that $\left(\frac{-2}{p}\right) = 1$ if $p \equiv 1$ or $3 \mod 8$, and $\left(\frac{-2}{p}\right) = -1$ if $p \equiv 5$ or $7 \mod 8$.

4. Find $\left(\frac{91}{167}\right)$ using quadratic reciprocity.

5. Evaluate the following Legendre symbols:

   (a) $\left(\frac{11}{37}\right)$; (b) $\left(\frac{19}{31}\right)$; (c) $\left(\frac{97}{101}\right)$; (d) $\left(\frac{31}{167}\right)$; (e) $\left(\frac{5}{160465489}\right)$; (f) $\left(\frac{3083}{3911}\right)$; (g) $\left(\frac{43691}{65537}\right)$.

Unit Summary

In this unit we have discussed about field, finite fields, the existence of multiplicative generators of finite field, how to find the quadratic residues and reciprocity and the square root of a residue. Also, we have studied the Legendre symbol and its properties.

Glossary

Irreducible polynomial	- Polynomials that are not divisible by any polynomial of lower degree except for constants.
Root	- A value which gives 0 when substituted in place of x in a polynomial.
Primitive polynomial	- Irreducible monic polynomial having exactly one root in the field.
Guassian integers	- Complex numbers whose real and imaginary parts are integers.

Exercise 3.

1. Suppose that f is a power of a prime $\updownarrow$. Find a simple formula for the number of monic irreducible polynomials of degree f over $\mathbf{F}_p$.

2. By computing $g.c.d.(f, f')$, find all multiple roots of $f(X) = X^7 + X^5 + X^4 - X^3 - X^2 - X + 1 \in \mathbf{F}_3[X]$ in its splitting field.

3. Suppose that $\alpha \in \mathbf{F}_{p^2}$ satisfies the polynomial $X^2 + aX + b$, where $a, b \in \mathbf{F}_p$.

   (a) Prove that α^p also satisfies this polynomial.

   (b) Prove that if $\alpha \notin \mathbf{F}_p$, then $a = -\alpha - \alpha^p$ and $b = \alpha^{p+1}$.

   (c) Prove that if $\alpha \notin \mathbf{F}_p$ and $c, d \in \mathbf{F}_p$ then $(c\alpha+d)^{p+1} = d^2 - acd + bc^2$ (which is $\in \mathbf{F}_p$).

   (d) Let i be a square root of -1 in $\mathbf{F}_{19^2}$. Use part (c) to find $(2+3i)^{101}$ (i.e., write it in the form $a + bi$, $a, b \in \mathbf{F}_{19}$).

4. For each of the following fields $\mathbf{F}_q$, where $q = p^f$, find an irreducible polynomial with coefficients in the prime field whose root α is primitive (i.e., generates $\mathbf{F}_q^*$), and write all of the powers of α as polynomials in α of degree $< f$: (a) F_4; (b) F_8; (c) F_{27}; (d) F_{25}.

5. Find the Gauss sum $G = \sum_{j=1}^{q-1} \left(\frac{j}{q}\right)\xi^j$ (here ξ is a q-th root of 1 in $\mathbf{F}_{p^f}$, where $p^f \equiv 1 \mod q$) when:

   (a) $q = 7$, $p = 29$, $f = 1$, $\xi = 7$;

   (b) $q = 5$, $p = 19$, $f = 2$, $\xi = 2 - 4i$, where i is a root of $X^2 + 1$;

(c) $q = 7$, $p = 13$, $f = 2$, $\xi = 4 + \alpha$, where α is a root of $X^2 - 2$.

6. Let $m = a^4 + 1$, $a \leq 2$. Find a positive integer x between 0 and $m/2$ such that $x^2 \equiv 2 \mod m$. Use this to find $\sqrt{2}$ in $\mathbf{F}_p$ when p is each of the following: the Fermat primes 17, 257, 65537; $p = 41 = (3^4+1)/2$, $p = 1297$, and $p = 1201$. (Hint: see the proof of Proposition 3.3.9.)

7. Let p and q be two primes with $q \equiv 1 \mod p$. Let ξ be a primitive p-th root of unity in $\mathbf{F}_q$. Find a formula in terms of ξ for a square root of $\left(\frac{-1}{p}\right)p$ in $\mathbf{F}_q$.

8. Evaluate the Legendre symbol $\left(\frac{1801}{8191}\right)$ (a) using the reciprocity law only for the Legendre symbol (i.e., factoring all numbers that arise), and (b) without factoring any odd integers, instead using the reciprocity law for the Jacobi symbol.

9. (a) Let p be an odd prime. Prove that -3 is a residue in $\mathbf{F}_p$ if and only if $p \equiv 1 \mod 3$.

   (b) Prove that 3 is a quadratic nonresidue modulo any Mersenne prime greater than 3.

10. Prove that a quadratic residue can never be a generator of $\mathbf{F}_p^*$.

11. (a) Let p be an odd prime, and let a, b, c be integers with $p \mid a$. Prove that the number of solutions $x \in \{0, 1, 2, ..., p-1\}$ to the congruence $ax^2 + bx + c \equiv 0 \mod p$ is given by the formula $1 + \left(\frac{D}{p}\right)$, where

$D = b2 - 4ac$ is the discriminant.

(b) How many solutions in $\mathbf{F}_{83}$ are there to each of the following equations: (i) $x^2 + 1 = 0$; (ii) $x^2 + x + 1 = 0$; (iii) $x^2 + 21x - 11 = 0$; (iv) $x^2 + x + 21 = 0$; (v) $x^2 - 4x - 13 = 0$?

(c) How many solutions in $\mathbf{F}_{97}$ are there to each of the equations in part (b)?

12. Let $p = 2081$, and let n be the smallest positive nonresidue modulo p. Find n, and use the method in the text to find a square root of 302 modulo p.

Answers :

Check your progress 3.1.

1. $x^2 + 2x + 2$

   $x^2 + x + 2$.

2. $Q(\sqrt{2}, i)$.

3. $\pi + 2$ is not algebraic over Q.

Check your progress 3.2.

1.

prime p	2	3	5	7	11	13	17
smallest generator	1	2	2	3	2	2	3
number of generators	1	1	2	2	4	4	8

2. 5^6.

3. 2 for $d = 1$: $X, X+1$; 1 for $d = 2$: X^2+X+1; 2 for $d = 3$: X^3+X^2+1, X^3+X+1; 3 for $d = 4$: X^4+X^3+1, X^4+X+1, $X^4+X^3+X^2+X+1$; 6 for $d = 5$: X^5+X^3+1, X^5+X^2+1, $X^5+X^4+X^3+X^2+1$, $X^5+X^4+X^3+X+1$, $X^5+X^4+X^2+X+1$, $X^5+X^3+X^2+X+1$; 9 for $d = 6$: X^6+X^5+1, X^6+X^3+1, X^6+X+1, $X^6+X^5+X^4+X^2+X+1$, $X^6+X^5+X^4+X+1$, $X^6+X^5+X^3+X^2+1$, $X^6+X^5+X^2+X+1$, $X^6+X^4+X^3+X+1$, $X^6+X^4+X^2+X+1$.

4. 3 for $d = 1$: X, $X \pm 1$; 3 for $d = 2$: X^2+1, $X^2 \pm X - 1$; 8 for$d = 3$: $X^3+X^2 \pm (X-1)$, $X^3-X^2 \pm (X+1)$, $X^3 \pm (X^2-1)$, $X^3-X \pm 1$; 18 for $d = 4$; 48 for $d = 5$; 116 for $d = 6$.

5. (a) $g.c.d. = 1 = X^2g + (X+1)f$;
   (b) $g.c.d. = X^3+X^2+1 = f + (X^2+X)g$;
   (c) $g.c.d. = 1 = (X-1)f - (X^2-X+1)g$;
   (d) $g.c.d. = X+1 = (X-1)f - (X^3-X^2+1)g$;
   (e) $g.c.d. = X+78 = (50X+20)f + (51X^3+26X^2+27X+4)g$.

Check your progress 3.3.

1. The sets of residues are: for $p = 3$, $\{1\}$; for $p = 5$, $\{1, 4\}$; for $p = 7$, $\{1, 2, 4\}$; for $p = 13$, $\{1, 3, 4, 9, 10, 12\}$; for $p = 17$, $\{1, 2, 4, 8, 9, 13, 15, 16\}$; for $p = 19$, $\{1, 4, 5, 6, 7, 9, 11, 16, 17\}$.

2. $g.c.d.(84, 1330) = 14$.

3. Write $\left(\frac{-2}{p}\right) = \left(\frac{-1}{p}\right)\left(\frac{2}{p}\right)$, and consider the four possible cases of p mod 8.

4. $\left(\frac{91}{167}\right) = \left(\frac{7}{167}\right)\left(\frac{13}{167}\right) = -\left(\frac{167}{7}\right)\left(\frac{167}{13}\right) = -\left(\frac{-1}{7}\right)\left(\frac{-2}{13}\right) = -(-1)(-1) = -1.$

5. (a) 1; (b) 1; (c) 1; (d) 1; (e) 1; (f) 1; (g) -1.

Exercise 3.

1. $(p^f - p^{f/\updownarrow})/f$.

2. Since $g.c.d.(f, f') = X^2 + 1$, the multiple roots are $\pm\alpha^2$, where α is the generator of $\mathbf{F}_9^*$ in the text.

3. (a) Raising $0 = \alpha^2 + b\alpha + c$ to the p-th power and using the fact that $b^p = b$ and $c^p = c$, we obtain $0 = (\alpha^p)^2 + b\alpha^p + c$. (b) The polynomial's two distinct roots are then α and α^p. Then a is minus the sum of the roots, and b is the product of the roots. (c) $(c\alpha+d)^{p+1} = (c\alpha^p + d)(c\alpha + d)$, and then multiply out and use part (b). (d) $(2 + 3i)^{5(19+1)+1} = (2^2 + 3^2)^5(2 + 3i) = 14(2 + 3i) = 9 + 4i$.

4. (a) Let α be a root of $X^2+X+1 = 0$; then the-three successive powers of α are α, $\alpha + 1$, and 1. (b) Let α be a root of $X^3+X+1 = 0$; then the seven successive powers of α are α, α^2, $\alpha + 1$, $\alpha^2 + \alpha$, $\alpha^2+\alpha+1$, $\alpha^2 + 1$, 1. (c) Let α be a root of $X^3 - X - 1 = 0$; then the 26 successive powers of α are α, α^2, α+l, $\alpha^2+\alpha$, $\alpha^2+\alpha+1$, $\alpha^2-\alpha+1$, $-\alpha^2 - \alpha + 1$, $\alpha^2 - 1$, $-\alpha + 1$, $-\alpha^2 + \alpha$, $\alpha^2 - \alpha - 1$, $-\alpha^2 + 1$, -1,

followed by the same 13 elements with all $+'$s and $-'$s reversed. (d) Let α be a root of $X^2 - X + 2 = 0$; then the 24 successive powers of α are α, $\alpha - 2$, $-\alpha - 2$, $2\alpha + 2$, $-\alpha + 1$, 2, then the same six elements multiplied by 2, then multiplied by -1, then multiplied by -2, giving all 24 powers of α.

5. (a) 14; (b) 9; (c) 9α.

6. $a^3 - a$ (see the proof of Proposition 3.3.9); 6, 60, 4080, 24, 210, 336.

7. Since $q \equiv 1 \mod p$, there is a primitive p-th root of unity ξ in $\mathbf{F}_q$. Then $G = \sum_{j=1}^{p-1} \left(\frac{j}{p}\right)\xi^j$ has square $\left(\frac{-1}{p}\right)p$ (see the Lemma 3.3.12).

8. (a)$\left(\frac{1801}{8191}\right) = \left(\frac{8191}{1801}\right) = \left(\frac{987}{1801}\right) = \left(\frac{3}{1801}\right)\left(\frac{7}{1801}\right)\left(\frac{47}{1801}\right) = \left(\frac{1}{3}\right)\left(\frac{2}{7}\right)\left(\frac{15}{47}\right) = 1 \cdot 1 \cdot \left(\frac{3}{47}\right)\left(\frac{5}{47}\right) = -\left(\frac{2}{3}\right)\left(\frac{2}{5}\right) = -1.$
(b) $\left(\frac{987}{1801}\right) = \left(\frac{1801}{987}\right) = \left(\frac{2.407}{987}\right) = -\left(-1\right)\left(\frac{987}{407}\right) = \left(\frac{173}{407}\right) = \left(\frac{407}{173}\right) = \left(\frac{61}{173}\right) = \left(\frac{173}{61}\right) = \left(\frac{51}{61}\right) = \left(\frac{61}{51}\right) = \left(\frac{2.5}{51}\right) = -\left(\frac{5}{51}\right) = -\left(\frac{51}{5}\right) = -1.$

9. (a) $\left(\frac{-3}{p}\right) = \left(\frac{-1}{p}\right)\left(\frac{3}{p}\right) = \left(-1\right)^{(p-1/2)}\left(-1\right)^{(3-1)(p-1)/4}\left(\frac{p}{3}\right) = \left(\frac{p}{3}\right)$, which $= 1$ if and only if $p \equiv 1 \mod 3$.
(b) $\left(\frac{3}{2^p-1}\right) = -\left(\frac{2^p-1}{3}\right) = -\left(\frac{1}{3}\right) = -1.$

10. Any power of a residue is a residue, so none of the nonresidues can occur as a power, and that means a residue cannot be a generator.

11. (a) Solve by completing the square; show that the number of solutions is the same as for the equation $x2 \equiv D \mod p$. There is 1 solution if $D = 0$, none if D is a nonresidue, and 2 if D is a residue. (b) 0, 0, 2, 1, 2; (c) 2, 2, 1, 0, 0.

12. $n = 3$; $p - 1 = 2^5 \cdot 65$; $r \equiv a^{33} \equiv 203 \mod p$ (we compute 302^{33} by the repeated squaring method, successively squaring 5 times and multiplying the result by 302); also by the repeated squaring method we compute $b \equiv n^{65} \equiv 888 \mod p$; one takes $j = 2^2$, i.e., $\sqrt{302} \mod p \equiv b^4 r \equiv 1292 \mod p$.

References:

1. Neal Koblit, A course in Number Theory and Cryptography, Springer - Verlag, New York, 2nd edition, 2002.

Suggested Reading:

1. I. Niven and H. S. Zuckermann, An Introduction to Theory of Numbers (Edition 3), Wiley Eastern Ltd, New Delhi 1976

2. D. M. Burton, Elementary Number Theory, Brown Publishers, Iowa, 1989

3. K. Ireland and M. Rosen, A classic Introduction to Modern Number Theory, Springer - Verlag, 1972

4. N. Koblit, Algebraic Aspects of Cryptography, Springer-Verlag, 1998.

UNIT - 4

Unit 4

Cryptography

Objectives.

By studying this unit, the students will

1. understand the cryptosystem.
2. know enciphering matrices.
3. review linear algebra.
4. solve a system of simultaneous congruences.
5. know to encipher a plaintext and decipher a ciphertext.

4.1 Some simple cryptosystems.

Cryptosystem: Cryptography is the study of methods of sending messages in disguised form so that only the intended recipients can remove the disguise and read the message. The message we want to send is called the **plaintext** and the disguised message is called the **ciphertext**. The plaintext and ciphertext are written in some **alphabet** (usually, but not

always, they are written in the same alphabet) consisting of a certain number N of **letters**. The term "letter" (or "character") can refer not only to the familiar A-Z, but also to numerals, blanks, punctuation marks, or any other symbols that we allow ourselves to use when writing the messages. (If we don't include a blank, for example, then all of the words are run together, and the messages are harder to read.) The process of converting a plaintext to a ciphertext is called **enciphering** or **encryption**, and the reverse process is called **deciphering** or **decryption**.

The plaintext and ciphertext are broken up into message units. A message unit might be a single letter, a pair of letters (**digraph**), a triple of letters (**trigraph**), or a block of 50 letters. An **enciphering transformation** is a function that takes any plaintext message unit and gives us a ciphertext message unit. In other words, it is a map f from the set $\mathcal{P}$ of all possible plaintext message units to the set $\mathcal{C}$ of all possible ciphertext message units. We shall always assume that f is a 1-to-1 correspondence. That is, given a ciphertext message unit, there is one and only one plaintext message unit for which it is the encryption. The **deciphering transformation** is the map f^{-1} which goes back and recovers the plaintext from the ciphertext. We can represent the situation schematically by the diagram

$$\mathcal{P} \xrightarrow{f} \mathcal{C} \xrightarrow{f^{-1}} \mathcal{P}$$

Any such set-up is called a **cryptosystem**.

The first step in inventing a cryptosystem is to "label" all possible

plaintext message units and all possible ciphertext message units by means of mathematical objects from which functions can be easily constructed. These objects are often simply the integers in some range. For example, if our plaintext and ciphertext message units are single letters from the 26-letter alphabet A-Z, then we can label the letters using the integers 0, 1, 2, $\cdots$, 25, which we call their "numerical equivalents." Thus, in place of A we write 0, in place of S we write 18, in place of X we write 23, and so on. As another example, if our message units are digraphs in the 27-letter alphabet consisting of A-Z and a blank, we might first let the blank have numerical equivalent 26 (one beyond Z), and then label the digraph whose two letters correspond to $x, y \in \{0, 1, 2, \cdots, 26\}$ by the integer

$$27x + y \in \{0, 1, \cdots, 728\}.$$

Thus, we view the individual letters as digits to the base 27 and we view the digraph as a 2-digit integer to that base. For example, the digraph "NO" corresponds to the integer $27 \cdot 13 + 14 = 365$. Analogously, if we were using trigraphs as our message units, we could label them by integers $729x + 27y + z \in \{0, 1, \cdots, 19682\}$. In general, we can label blocks of k letters in an N-letter alphabet by integers between 0 and $N^k - 1$ by regarding each such block as a k-digit integer to the base N.

In some situations, one might want to label message units using other mathematical objects besides integers - for example, vectors or points on some curve. But for the duration of this section we shall use integers.

Remark 4.1.1. Let us start with the case when we take a message unit (of plaintext or of ciphertext) to be a single letter in an N-letter alphabet labeled by the integers $0, 1, 2, \cdots, N-1$. Then, by definition, an enciphering transformation is a rearrangement of these N integers.

To facilitate rapid enciphering and deciphering, it is convenient to have a relatively simple rule for performing such a rearrangement. One way is to think of the set of integers $\{0, 1, 2, \cdots, N-1\}$ as $\mathbf{Z}/N\mathbf{Z}$, and make use of the operations of addition and multiplication modulo N.

Example 4.1.2. Suppose we are using the 26-letter alphabet A - Z with numerical equivalents 0 - 25. Let the letter $P \in \{0, 1, \cdots, 25\}$ stand for a plaintext message unit. Define a function f from the set $\{0, 1, \cdots, 25\}$ to itself by the rule

$$f(P) = \begin{cases} P + 3, & if\ x < 23 \\ P - 23, & if\ x \geq 23 \end{cases}$$

In other words, f simply adds 3 modulo 26: $f(P) \equiv P + 3 \mod 26$. The definition using modular arithmetic is easier to write down and work with. Thus, with this system, to encipher the word "YES" we first convert to numbers: 24 4 18, then add 3 moduo 26: 1 7 21, then translate back to letters: "BHV".To decipher a message, one subtracts 3 modulo 26. For example, the ciphertext "ZKB" yields the plaintext "WHY". This

cryptosystem was apparently used in ancient Rome by Julius Caesar, who supposedly invented it himself.

Remark 4.1.3. Example 4.1.2 can be generalized as follows. Suppose we are using an N-letter alphabet with numerical equivalents $0, 1, \cdots, N-1$. Let b be a fixed integer. By a **shift** transformation we mean the enciphering function f defined by the rule $C = f(P) \equiv P + b \mod N$. Julius Caesar's cryptosystem was the case $N = 26$, $b = 3$. To decipher a ciphertext message unit $C \in \{0, 1, \cdots, N-1\}$, we simply compute $P = f^{-1}(C) \equiv C - b \mod N$.

Cryptanalysis. Now suppose that you are not privy to the enciphering and deciphering information, but you would nevertheless like to be able to read the coded messages. This is called **breaking** the code, and the science of breaking codes is called **cryptanalysis**.

In order to break a cryptosystem, one needs two types of information. The first is the general nature (the **structure**) of the system. For example, suppose we know that the cryptosystem uses a shift transformation on single letters of the 26-letter alphabet A-Z with numerical equivalents 0-25, respectively. The second type of information is knowledge of a specific choice of certain parameters connected with the given type of crptosystem. In our example, the second type of information one needs to know is the choice of the shift parameter b. Once one has that information, one can encipher and decipher by the formulas $C \equiv P + b \mod N$ and

$P \equiv C - b \mod N$.

We shall always assume that the general structural information is already known. In practice, users of cryptography often have equipment for enciphering and deciphering which is constructed to implement only one type of cryptosystem. Over a period of time the information about what type of system they're using might leak out. To increase their security, therefore, they frequently change the choice of parameters used with the system. For example, suppose that two users of the shift cryptosystem are able to meet once a year. At that time they agree on a list of 52 choices of the parameter b, one for each week of the coming year.

The parameter b (more complicated cryptosystems usually have several parameters) is called a **key**, or, more precisely, the **enciphering key**.

Example 4.1.4. So suppose that we intercept the message "FQOCUDEM", which we know was enciphered using a shift transformation on single letters of the 26-letter alphabet, as in the Example 4.1.2 . It remains for us to find the b. One way to do this is by **frequency analysis**. This works as follows. Suppose that we have already intercepted a long string of ciphertext, say several hundred letters. We know that "E" is the most frequently occurring letter in English language. So it is reasonable to assume that the most frequently occurring letter in the ciphertext is the encryption of E. Suppose that we find that "U" is the most frequently occurring character in the ciphertext. That means that the shift takes "E" = 4 to "U" = 20, i.e., $20 \equiv 4 + b \mod 26$, so that $b = 16$. To deci-

pher the message, then, it remains for us to subtract 16 (working modulo 26) from the numerical equivalents of "FQOCUDEM":
"FQOCUDEM" = 5 16 14 2 20 3 4 12 $\mapsto$ 15 0 24 12 4 13 14 22 = "PAYMENOW".

Remark 4.1.5. In the case of the shift encryption of single letters of a 26-letter alphabet it is not even necessary to have a long string of ciphertext to find the most frequently occurring letter. After all, there are only 26 possibilities for b, and one can simply run through all of them. Most likely, only one will give a message that makes any sense, and that b is the enciphering key.

Thus, this type of cryptosystem is too simple to be much good. It is too easy to break. An improvement is to use a more general type of transformation of $\mathbf{Z}/N\mathbf{Z}$, called an **affine map**: $C \equiv aP + b \mod N$, where a and b are fixed integers (together they form the enciphering key). For example, working again in the 26-letter alphabet, if we want to encipher our message "PAYMENOW" using the affine transformation with enciphering key a=7, b= 12, we obtain: 15 0 24 12 4 13 14 22 $\mapsto$ 13 12 24 18 14 25 6 10 = "NMYSOZGK".

To decipher a message that was enciphered by means of the affine map $C \equiv aP + b \mod N$, one simply solves for P in terms of C, obtaining $P \equiv a'C + b' \mod N$, where a' is the inverse of a modulo N and b' is equal to $-a^{-1}b$. Note that this works ony if $g.c.d.(a, N) = 1$; otherwise we cannot solve for P in terms of C. If $g.c.d(a, N) > 1$, then it is easy

to see that more than one plaintext letter will give the same ciphertext letter, so we cannot uniquely recover the plaintext from the ciphertext. By definition, that is not an enciphering transformation: we always require that the map be 1-to-1, i.e., that the plaintext be uniquely determined from the ciphertext. To summarize, an affine cryptosystem in an N-letter alphabet with parameters $a \in \left(\mathbf{Z}/N\mathbf{Z}\right)^*$ an $b \in \mathbf{Z}/N\mathbf{Z}$ consists of the rules:

$$C \equiv aP + b \mod N,\ P \equiv a'C + b' \mod N,$$

where

$$a' = a^{-1} \text{ in } \left(\mathbf{Z}/N\mathbf{Z}\right)^*,\ b' = -a^{-1}b.$$

As a special case of the affine cryptosystems we can set a=1, thereby obtaining the shift transformations. Another special case is when b=0: $P \equiv aC \mod N$, $C \equiv a^{-1}P \mod N$. The case b=0 is called a **linear** transformation, meaning that the map takes a sum to a sum, i.e., if C_1 is the encryption of P_1 and C_2 is the encryption of P_2, then $C_1 + C_2$ is a encryption of $P_1 + P_2$ (where, of course, we are adding modulo N).

Now suppose that we know that an intercepted message was enciphered using an affine map single letters in an N-letter alphabet. We would like to determine the enciphering key a, b so that we can read the message. We need two bits of information to do this.

Example 4.1.6. Still working in our 26-letter alphabet, suppose that we know the most frequently occurring letter of ciphertext is "K", and the second most frequently occurring letter is "D". It is reasonable to assume that these are the encryptions of "E" and "T", respectively, which are the two most frequently occurring letters in the English language. Thus, replacing the letters by their numerical equivalents and substituting for P and C in the deciphering formula, we obtain:

$$10a' + b' \equiv 4 \mod 26,$$
$$3a' + b' \equiv 19 \mod 26.$$

We have two congruences with two unknowns, a' and b'. The quickest way to solve is to subtract the two congruences to eliminate b'. We obtain $7a' \equiv 11 \mod 26$, and $a' \equiv 7^{-1}11 \equiv 9 \mod 26$. Finally, we obtain b' by substituting this value for a' in one of the congruences: $b' \equiv 4 - 10a' \equiv 18 \mod 26$. So messages can be deciphered by means of the formula $P \equiv 9C + 18 \mod 26$.

Remark 4.1.7. Recall from linear algebra that n equations suffice to find n unknowns only if the equations are independent (i.e., if the determinant is nonzero). For example, in the case of 2 equations in 2 unknowns this means that the straight line graphs of the equations intersect in a single point (are not parallel). In our situation, when we try to cryptanalyze an affine system from the knowledge of the two most frequently occurring letters of ciphertext, we might find that we cannot solve the two

congruences uniquely for a' and b'.

Example 4.1.8. Suppose that we have a string of ciphertext which we know was enciphered using an affine transformation of single letters in a 28-letter alphabet consisting of A-Z, a blank, and ?, where A-Z have numerical equivalents 0-25, blank=26, ?=27. A frequency analysis reveals that the two most common letters of ciphertext are "B" and "?", in that order. Since the most common letters in an English language text written in this 28-letter alphabet are " " (blank) and "E", in that order, we suppose that "B" is the encryption of " " and "?" is the encryption of "E". This leads to the two congruences: $a' + b' \equiv 26 \mod 28$, $27a' + b' \equiv 4 \mod 28$. Subtracting the two congruences, we obtain: $2a' \equiv 22 \mod 28$, which is equivalent to the congruence $a' \equiv 11 \mod 14$. This means that $a' \equiv 11$ *or* $25 \mod 28$, and then $b' \equiv 15$ *or* $1 \mod 28$, respectively. The fact of the matter is that both of the possible affine deciphering transformations $11C + 15$ and $25C + 1$ give " " and "E" as the plaintext letters corresponding to "B" and "?", respectively. At this point we could try both possibilities, and see which gives an intelligible message. Or we could continue our frequency analysis. Suppose we find that "I" is the third most frequently occurring letter of ciphertext. Using the fact that "T" is the third most common letter in the English language (of our 28 letters), we obtain a third congruence: $8a' + b' \equiv 19 \mod 28$. This extra bit of information is enough to determine which of the affine maps is the right one. We find that it is $11C + 15$.

Digraph transformation.

We now suppose that our plaintext and ciphertext message units are **two-letter** blocks, called **digraphs**. This means that the plaintext is split up into two-letter segments. If the entire plaintext has an odd number of letters, then in order to obtain a whole number of digraphs we add on an extra letter at the end; we choose a letter which is not likely to cause confusion, such as a blank if our alphabet contains a blank, or else "X" or "Q" if we are using just the 26-letter alphabet.

Each digraph is then assigned a numerical equivalent. The simplest way to do this is to take $xN + y$, where x is the numerical equivalent of the first letter in the digraph, y is the numerical equivalent of the second letter in the digraph, and N is the number of letters in the alphabet. Equivalently, we think of a digraph as a 2-digit base-N integer. This gives a 1-to-1 correspondence between the set of all digraphs in the N-letter alphabet and the set of all nonnegative integers less than N^2. We described this "labeling" of digraphs before in the special case when $N = 27$.

Next, we decide upon an enciphering transformation, i.e., a rearrangement of the integers $\{0, 1, 2, \cdots, N^2-1\}$. Among the simplest enciphering transformations are the **affine** ones, where we view this set of integers as $\mathbf{Z}/N^2\mathbf{Z}$, and define the encryption of P to be the nonnegative integer less than N^2 satisfying the congruence $C \equiv aP + b \mod N^2$. Here, as before, a must have no common factor with N (which means it has no common

factor with N^2), in order that we have an inverse transformation telling us how to decipher: $P \equiv a'C + b' \mod N^2$, where $a' \equiv a^{-1} \mod N^2$, $b' \equiv -a^{-1}b \mod N^2$. We translate C into a two-letter block of ciphertext by writing it in the form $C = x'N + y'$, and then looking up the letters with numerical equivalents x' and y'.

Example 4.1.9. Suppose we are working in the 26-letter alphabet and using the digraph enciphering transformation $C \equiv 159P + 580 \mod 676$. Then the digraph "NO" has numerical equivalent $13 \cdot 26 + 14 = 352$ and is taken to the ciphertext digraph $159 \cdot 352 + 580 \equiv 440 \mod 676$, which is "QY". The digraph "ON" has numerical equivalent 377, and is taken to 359="NV". Notice that the digraphs change as a unit, and there is no relation between the encryption of one digraph and that of another one that has a letter in common with it or even consists of the same letters in the reverse order.

Remark 4.1.10. To break a digraphic encryption system which uses an affine transformation $C \equiv aP + b \mod N^2$, we need to know the ciphertext corresponding to two different plaintext message units. Since the message units are digraphs, a frequency analysis meas counting which two-letter blocks occur most often in a long string of ciphertext (of course, counting only those occurrences where the first letter begins a message unit, ignoring the occurrences of the two letters which straddle two message units), and comparing with the known frequency of digraphs in English language texts (written in the same alphabet). For example, if we use the 26-letter alphabet, statistical analyses seem to show that "TH"

and "HE" are the two most frequently occurring digraphs, in that order. Knowing two plaintext-ciphertext pairs of digraphs is often (but not always) enough to determine a and b.

Example 4.1.11. You know that your adversary is using a cryptosystem with a 27-letter alphabet, in which the letters A-Z have numerical equivalents 0-25, and blank=26. Each digraph then corresponds to an integer between 0 and $728 = 27^2 - 1$ according to the rule that, if the two letters in the digraph have numerical equivalents x and y, then the digraph has numerical equivalent $27x + y$, as explained earlier. Suppose that a study of a large sample of ciphertext reveals that the most frequently occurring digraphs are (in order) "ZA","IA",and "IW". Suppose that the most common digraphs in the English language (for text written in our 27-letter alphabet) are "E " (i.e., "E blank"), "S "," T". You know that the cryptosystem uses an affine enciphering transformation modulo 729. Find the deciphering key, and read the message "NDXBHO". Also find the enciphering key.

Solution. We know that plaintexts are enciphered by means of the rule $C \equiv aP + b \mod 729$, and that ciphertexts can be deciphered by means of the rule $P \equiv a'C + b' \mod 729$; here a, b form the enciphering key, and a', b' form the deciphering key. We first want to find a' and b'. We know how three digraphs are deciphered, and, after we replace the digraphs by their numerical equivalents, this gives us the three congruences:

$$675a' + b' \equiv 134 \mod 729,$$
$$216a' + b' \equiv 512 \mod 729,$$
$$238a' + b' \equiv 721 \mod 729.$$

If we try to eliminate b' by subtracting the first two congruences, we arrive at $459a' = 351 \mod 729$, which does not have a unique solution a' mod 729 (there are 27 solutions). We do better if we subtract the third congruence from the first, obtaining $437a' \equiv 142 \mod 729$. To solve this, we must find the inverse of 437 modulo 729. By way of review of the Euclidean algorithm, let's go through that in detail:

$$729 = 437 + 292$$
$$437 = 292 + 145$$
$$292 = 2 \cdot 145 + 2$$
$$145 = 72 \cdot 2 + 1$$

and then

$$\begin{aligned} 1 &= 145 - 72 \cdot 2 \\ &= 145 - 72(292 - 2 \cdot 145) \\ &= 145 \cdot 145 - 72 \cdot 292 \\ &= 145(437 - 292) - 72 \cdot 292 \\ &= 145 \cdot 437 - 217 \cdot 292 \\ &= 145 \cdot 437 - 217(729 - 437) \\ &\equiv 362 \cdot 437 \mod 729. \end{aligned}$$

Thus, $a' \equiv 362 \cdot 142 \equiv 374 \mod 729$, and then $b' \equiv 134 - 675 \cdot 374 \equiv 647$ mod 729. Now applying the deciphering transformation to the digraphs "ND", "XB" and "HO" of our message - they correspond to the integers 354, 622 and 203, respectively - we obtain the integers 365, 724 and 24. Writing $365 = 13 \cdot 27 + 14$, $724 = 26 \cdot 27 + 22$, $24 = 0 \cdot 27 + 24$, we put together the plaintext digraphs into the message "NO WAY". Finally, to find the enciphering key we compute $a \equiv a'^{-1} \equiv 374^{-1} \equiv 614 \mod 729$ (again using the Euclidean algorithm) and $b \equiv -a'^{-1}b' \equiv -614 \cdot 647 \equiv 47$ mod 729.

Remark 4.1.12. Although affine cryptosystems with digraphs (i.e., modulo N^2) are better than the ones using single letters (i.e., modulo N), they also have drawbacks. Notice that the second letter of each ciphertext digraph depends only on the second letter of the plaintext digraph. This is because that second letter depends on the mod$N$ value of $C \equiv aP + b$ mod N^2, which depends only on P modulo N, i.e., only on the second letter of the plaintext digraph. Thus, one could obtain a lot of information (namely, a and b modulo N) from a frequency analysis of the even-numbered letters of the ciphertext message. A similar remark applies to mod-N^k affine transformations of k-letter blocks.

Let Us Sum Up

- Cryptography is the study of methods of sending messages in disguised form so that only the intended recipients can remove the disguise and read the message.
- The message we want to send is called the plaintext.
- The disguised message is called the ciphertext.
- The process of converting a plaintext to a ciphertext is called enciphering or encryption, and reverse process is called deciphering or decryption.
- An enciphering transformation is a function that takes any plaintext message unit and gives us a ciphertext message unit.
- The deciphering transformation is the map f^{-1} which goes back and recovers the plaintext from the ciphertext.
- The science of breaking codes is called cryptanalysis.

Check your progress 4.1.

1. In the 27-letter alphabet (with blank=26), use the affine enciphering transformation with key $a = 13$, $b = 9$ to encipher the message "HELP ME."
2. How many different shift transformations are there with an N-letter alphabet?

3. Find a formula for the number of different affine enciphering transformations there are with an N-letter alphabet.

4. How many affine transformations are there when $N = 26$, 27, 29, 30?

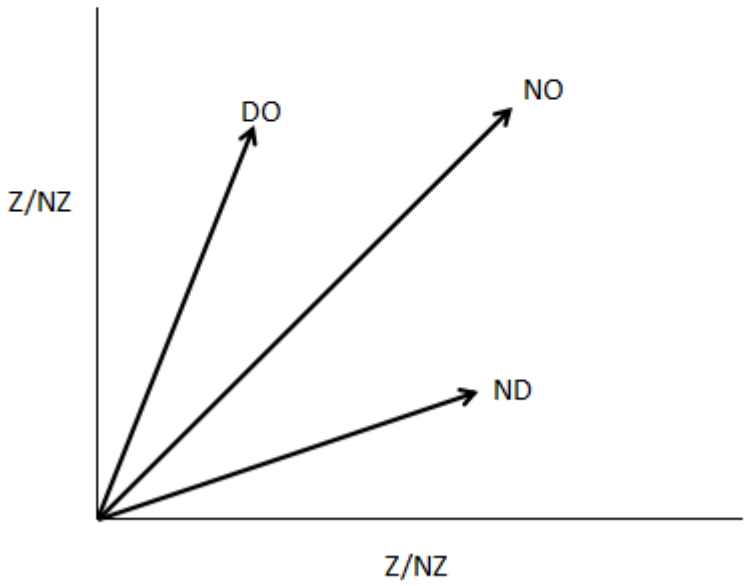
4.2 Enciphering Matrices

Suppose we have an N-letter alphabet and want to send digraphs (two-letter blocks) as our message units. In section 4.1 we saw how we can let each digraph correspond to an integer considered modulo N^2, i.e., to an element of $\mathbf{Z}/N^2\mathbf{Z}$. An alternate possibility is to let each digraph correspond to a vector, i.e., to a pair of integers $\begin{pmatrix} x \\ y \end{pmatrix}$ with x and y each considered modulo N. For example, if we're using the 26-letter alphabet A-Z with numerical equivalents 0-25, respectively, then the digraph NO corresponds to the vector $\begin{pmatrix} 13 \\ 14 \end{pmatrix}$. See the diagram.

We picture each digraph P as a point on an $N \times N$ square array. That is, we have an "xy-plane", except that each axis, rather than being a copy of the real number line, is now a copy of $\mathbf{Z}/N\mathbf{Z}$. Just as the real xy-plane is often denoted R^2, this $N \times N$ array is denoted $\left(\mathbf{Z}/N\mathbf{Z}\right)^2$.

Once we visualize digraphs as vectors (points in the plane), we then interpret an "enciphering transformation" as a rearrangement of the $N \times N$ array of points. More precisely, an enciphering map is a 1-to-l function from $\left(\mathbf{Z}/N\mathbf{Z}\right)^2$ to itself.

Remark 4.2.1. For several centuries one of the most popular methods



of encryption was the so-called "Vigenére cipher." This can be described as follows. For some fixed k, regard blocks of k letters as vectors in $\left(\mathbf{Z}/N\mathbf{Z}\right)^k$. Choose some fixed vector $b \in \left(\mathbf{Z}/N\mathbf{Z}\right)^k$ (usually b was the vector corresponding to some easily remembered "key-word"), and encipher by means of the vector translation $C = P + b$ (where the ciphertext message unit C and the plaintext message unit P are k-tuples of integers modulo N). This cryptosystem, unfortunately, is almost as easy to break as a single-letter translation (see Example 4.1.1). Namely, if one knows (or can guess) N and k, then one simply breaks up the ciphertext in blocks of k letters and performs a frequency analysis on the first letter in each block to determine the first component of b, then the same for the

second letter in each block, and so on.

Review of linear algebra. We now review how one works with vectors in the real xy-plane and with 2×2-matrices with real entries. Recall that, given a 2×2 array of numbers

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \text{ and a vector in the plane } \begin{pmatrix} x \\ y \end{pmatrix}$$

(we shall write vectors as columns), one can *apply the matrix to the vector* to obtain a new vector, as follows:

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} =_{def} \begin{pmatrix} ax + by \\ cx + dy \end{pmatrix}.$$

For a fixed matrix, this function from one vector to another vector is called a linear transformation, meaning that it preserves sums and constant multiples of vectors. Using this notation, we can view any set of simultaneous equations of the form $ax + by = e$, $cx + dy = f$ as equivalent to a single matrix equation $AX = B$, where A denotes the matrix

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix},$$

X denotes the vector of unknowns $\begin{pmatrix} x \\ y \end{pmatrix}$, and B denotes the vector of constants $\begin{pmatrix} e \\ f \end{pmatrix}$. Stated in words, the simultaneous equations can thus be interpreted as asking to find a vector which when "multiplied" by a certain known matrix gives a certain known vector. Thus, it is analogous

to the simple equation $ax = b$, which is solved by multiplying both sides by a^{-1} (assuming $a \neq 0$). Similarly, one way to solve the matrix equation $AX = B$ is to find the inverse of the matrix A, and then apply A^{-1} to both sides to obtain the unique vector solution $X = A^{-1}B$.

By the inverse of the matrix A we mean the matrix which multiplies by it to give the identity matrix

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

(the matrix which, when applied to any vector, keeps that vector the same). But not all matrices have inverses. It is not hard to prove that a matrix

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

has an inverse if and only if its determinant $D =_{def} ad - bc$ is nonzero, and that its inverse in that case is

$$\frac{1}{D}\begin{pmatrix} d & -b \\ -c & a \end{pmatrix} = \begin{pmatrix} D^{-1}d & -D^{-1}b \\ -D^{-1}c & D^{-1}a \end{pmatrix}$$

There are three possibilities for the solutions of the system of simultaneous equations $AX = B$. First, if the determinant D is nonzero, then there is precisely one solution $X = \begin{pmatrix} x \\ y \end{pmatrix}$. If $D = 0$, then either there are no solutions or there are infinitely many. The three possibilities have a simple geometric interpretation. The two equations give straight lines in the xy-plane. If $D \neq 0$, then they intersect in exactly one point (x, y).

Otherwise, they are parallel lines, which means either that they don't meet at all (the simultaneous equations have no common solution) or else that they are really the same line (the equations have infinitely many common solutions).

Next, let us suppose that we have a bunch of vectors $X_1 = \begin{pmatrix} x_1 \\ y_1 \end{pmatrix}, \cdots,$ $X_k = \begin{pmatrix} x_k \\ y_k \end{pmatrix}$, arranged as the columns of a $2 \times k$-matrix. Then we define the matrix product

$$AX = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x_1 & \cdots & x_k \\ y_1 & \cdots & y_k \end{pmatrix} =_{def} \begin{pmatrix} ax_1 + by_1 & \cdots & ax_k + by_k \\ cx_1 + dy_1 & \cdots & cx_k + dy_k \end{pmatrix},$$

i.e., we simply apply the matrix A to each column vector in order, obtaining new column vectors. For example, the product of two 2×2-matrices is:

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} a' & b' \\ c' & d' \end{pmatrix} = \begin{pmatrix} aa' + bc' & ab' + bd' \\ ca' + dc' & cb' + dd' \end{pmatrix}$$

Similar facts hold for 3×3-matrices, which can be applied to 3-dimensional column-vectors, and so on. However, the formulas for the determinant and inverse matrix are more complicated. This concludes our brief review of linear algebra over the real numbers.

Linear algebra modulo N. In section 4.1, when we were dealing with single characters and enciphering maps of $\mathbf{Z}/N\mathbf{Z}$, we found that two easy types of maps to work with were:

(a) "linear" maps $C = aP$, where a is invertible in $\mathbf{Z}/N\mathbf{Z}$;

(b) "affine" maps $C = aP + b$, where a is invertible in $\mathbf{Z}/N\mathbf{Z}$.

We have a similar situation when our message units are digraph-vectors. We first consider linear maps. The difference when we work with $\left(\mathbf{Z}/N\mathbf{Z}\right)^2$ rather than $\mathbf{Z}/N\mathbf{Z}$ is that now instead of an integer a we need a 2×2-matrix, which we shall denote A. We start by giving a systematic explanation of the type of matrices we need.

Let R be any commutative ring, i.e., a set with multiplication and addition satisfying the same rules as in a field, except that we do not require that any nonzero element have a multiplicative inverse. For example, $\mathbf{Z}/N\mathbf{Z}$ is always a ring, but it is not a field unless N is prime. We let R^* denote the subset of invertible elements of R. For example, $\left(\mathbf{Z}/N\mathbf{Z}\right)^* = \{0 < j < N | g.c.d.(j, N) = 1\}$

If R is a commutative ring, we let $M_2(R)$ denote the set of all 2×2- matrices with entries in R, with addition and multiplication defined in the usual way for matrices. We call $M_2(R)$ a "matrix ring over R" ; $M_2(R)$ itself is a ring, but it is not a commutative ring, i.e., in matrix multiplication the order of the factors makes a difference.

Earlier in this section, the matrices considered were the case when $R = \mathbf{R}$ is the ring (actually, field) of real numbers. Recall that a matrix

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

with real numbers a, b, c, d has a multiplicative inverse if and only if the

determinant $D = ad - bc$ is nonzero, and in that case the inverse matrix is

$$\begin{pmatrix} D^{-1}d & -D^{-1}b \\ -D^{-1}c & D^{-1}a \end{pmatrix}$$

We have a similar situation when we work over an arbitrary ring R.

Namely, suppose that

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in M_2(R)$$

and $D = det(A) =_{def} ad - bc$ is in R^*. Let D^{-1} denote the multiplicative inverse of D in R. Then

$$\begin{pmatrix} D^{-1}d & -D^{-1}b \\ -D^{-1}c & D^{-1}a \end{pmatrix} \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} D^{-1}(da - bc) & 0 \\ 0 & D^{-1}(-cb + ad) \end{pmatrix}$$
$$= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},$$

and we obtain the same result

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

if we multiply in the opposite order. Thus, A has an inverse matrix given by the same formula as in the real number case:

$$A^{-1} = \begin{pmatrix} D^{-1}d & -D^{-1}b \\ -D^{-1}c & D^{-1}a \end{pmatrix}$$

Example 4.2.2. Find the inverse of

$$A = \begin{pmatrix} 2 & 3 \\ 7 & 8 \end{pmatrix} \in M_2(\mathbf{Z}/26\mathbf{Z})$$

Solution. Here $D = 2{\cdot}8 - 3{\cdot}7 = -5 = 21$ in $\mathbf{Z}/26\mathbf{Z}$. Since $g.c.d.(21, 26) = 1$, the determinant D has an inverse, namely $21^{-1} = 5$. Thus,

$$A^{-1} = \begin{pmatrix} 5 \cdot 8 & -5 \cdot 3 \\ -5 \cdot 7 & 5 \cdot 2 \end{pmatrix} = \begin{pmatrix} 40 & -15 \\ -35 & 10 \end{pmatrix} = \begin{pmatrix} 14 & 11 \\ 17 & 10 \end{pmatrix}.$$

We check that $\begin{pmatrix} 14 & 11 \\ 17 & 10 \end{pmatrix} \begin{pmatrix} 2 & 3 \\ 7 & 8 \end{pmatrix} = \begin{pmatrix} 105 & 130 \\ 104 & 131 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$. Here, since we are working in $\mathbf{Z}/26\mathbf{Z}$, we are using "=" to mean that the entries are congruent modulo 26.

Just as in the real number case, a 2×2-matrix

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

with entries in a ring R can be multiplied by a column-vector $\begin{pmatrix} x \\ y \end{pmatrix}$ with $x, y \in R$ to get a new vector $\begin{pmatrix} x' \\ y' \end{pmatrix}$:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} ax + by \\ cx + dy \end{pmatrix}.$$

This gives a "linear map" from vectors to vectors, meaning that a linear combination $\begin{pmatrix} k_1x_1 + k_2x_2 \\ k_1y_1 + k_2y_2 \end{pmatrix}$, where k_1 and k_2 are in the ring R, is taken to $\begin{pmatrix} k_1x_1' + k_2x_2' \\ k_1y_1' + k_2y_2' \end{pmatrix}$. The only difference with the situation earlier in our review of linear algebra is that now everything is in our ring R rather than in the real numbers.

We shall want to apply all of this when our ring is $R = \mathbf{Z}/N\mathbf{Z}$. The next proposition will be stated in that case, although the analogous proposition is true for any R.

Proposition 4.2.3. *Let*

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in M_2(\mathbf{Z}/N\mathbf{Z})$$

and set $D = ad - bc$.

The following are equivalent:

(a) $g.c.d.(D, N) = 1$;

(b) A has an inverse matrix;

(c) if x *and* y *are not both 0 in* $(\mathbf{Z}/N\mathbf{Z})^2$, *then* $A \begin{pmatrix} x \\ y \end{pmatrix} \neq \begin{pmatrix} 0 \\ 0 \end{pmatrix}$;

(d) A gives a 1-to-1 correspondence of $\mathbf{Z}/N\mathbf{Z}^2$ *with itself.*

Proof. We already showed that (a)$\Longrightarrow$(b). It suffices now to prove that (b)$\Longrightarrow$(d)$\Longrightarrow$(c)$\Longrightarrow$(a).

Suppose that (b) holds. Then part (d) also holds, because A^{-1} gives the inverse map from $\begin{pmatrix} x' \\ y' \end{pmatrix}$ to $\begin{pmatrix} x \\ y \end{pmatrix}$. Next, if we have (d), then $\begin{pmatrix} x \\ y \end{pmatrix} \neq \begin{pmatrix} 0 \\ 0 \end{pmatrix}$ implies that $A \begin{pmatrix} x \\ y \end{pmatrix} \neq A \begin{pmatrix} 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$, and so (c) holds. Finally, we prove (c)$\Longrightarrow$(a) by showing that (a) false $\Longrightarrow$(c) false. So suppose that (a) is false, and set $m = g.c.d.(D, N) > 1$ and let $m' = N/m$. Three cases are possible.

Case (i). If all four entries of A are divisible by m, set $\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} m' \\ m' \end{pmatrix}$, to get a contradiction to (c).

Case (ii). If a and b are not both divisible by m, set $\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} -bm' \\ am' \end{pmatrix}$. Then

$$A \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} -bm' \\ am' \end{pmatrix} = \begin{pmatrix} -abm' + bam' \\ -cbm' + dam' \end{pmatrix} = \begin{pmatrix} 0 \\ Dm' \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix},$$

because $m|D$ and so $N = mm'|Dm'$.

Case (iii). If c and d are not both divisible by m, set $\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} dm' \\ -cm' \end{pmatrix}$, and proceed as in case (ii). These three cases exhaust all possibilities. Thus, (a) false implies (c) false. This completes the proof.

Example 4.2.4. Solve the following systems of simultaneous congruences:

$$\begin{aligned}
(a) \qquad & 2x + 3y \equiv 1 \mod 26, \\
& 7x + 8y \equiv 2 \mod 26; \\
(b) \qquad & x + 3y \equiv 1 \mod 26, \\
& 7x + 9y \equiv 2 \mod 26; \\
(c) \qquad & x + 3y \equiv 1 \mod 26, \\
& 7x + 9y \equiv 1 \mod 26.
\end{aligned}$$

Solution. The matrix form of the system (a) is $AX \equiv B \mod 26$, where $A = \begin{pmatrix} 2 & 3 \\ 7 & 8 \end{pmatrix} \in M_2(\mathbf{Z}/26\mathbf{Z})$, $X = \begin{pmatrix} x \\ y \end{pmatrix}$, and $B = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$. We obtain the unique solution

$$X \equiv A^{-1}B \equiv \begin{pmatrix} 14 & 11 \\ 17 & 10 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \end{pmatrix} \equiv \begin{pmatrix} 10 \\ 11 \end{pmatrix} \mod 26.$$

The matrix of the systems (b)-(c) does not have an inverse modulo 26, since its determinant is 14, which has a common factor of 2 with 26. However, we can work modulo 13, i.e., we can find the solution to the same congruence mod 13 and see if it gives a solution which works modulo 26. Modulo 13 we obtain

$$\begin{pmatrix} x \\ y \end{pmatrix} \equiv \begin{pmatrix} 9 & 10 \\ 6 & 1 \end{pmatrix} \begin{pmatrix} e \\ f \end{pmatrix}$$

(where $\begin{pmatrix} e \\ f \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$ in part (b) and $\begin{pmatrix} 1 \\ 1 \end{pmatrix}$ in part (c)). This gives $\begin{pmatrix} x \\ y \end{pmatrix} \equiv \begin{pmatrix} 3 \\ 8 \end{pmatrix}$ and $\begin{pmatrix} 6 \\ 7 \end{pmatrix}$ mod 13, respectively. Testing the possibilities modulo 26, we find that in part (b) there are *no* solutions, and in part (c) there are *two* solutions: $x = 6$, $y = 7$ and $x = 19$, $y = 20$.

Another way to solve systems of equations (preferable sometimes, especially when the matrix is not invertible) is to eliminate one of the variables (e.g., in parts (b) and (c), one could subtract 7 times the first congruence from the second).

Remark 4.2.5. To return to cryptography, we see from Proposition 4.2.3 that we can get enciphering transformations of our digraph-vectors by using matrices $A \in M_2(\mathbf{Z}/N\mathbf{Z})$ whose determinant has no common factor with N:

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}, \ D = ad - bc, \ g.c, d.(D, N) = 1.$$

Namely, each plaintext message unit $P = \begin{pmatrix} x \\ y \end{pmatrix}$ is taken to a ciphertext $C = \begin{pmatrix} x' \\ y' \end{pmatrix}$ by the rule

$$C = AP, \qquad \text{i.e.,} \ \begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}.$$

To decipher a message, we simply apply the inverse matrix:

$$P = A^{-1}AP = A^{-1}C, \qquad \text{i.e., } \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} D^{-1}d & -D^{-1}b \\ -D^{-1}c & D^{-1}a \end{pmatrix} \begin{pmatrix} x' \\ y' \end{pmatrix}$$

Example 4.2.6. Working in the 26-letter alphabet, use the matrix $A = \begin{pmatrix} 2 & 3 \\ 7 & 8 \end{pmatrix}$ to encipher the message unit "NO."

Solution. We have:

$$AP = \begin{pmatrix} 2 & 3 \\ 7 & 8 \end{pmatrix} \begin{pmatrix} 13 \\ 14 \end{pmatrix} = \begin{pmatrix} 68 \\ 203 \end{pmatrix} = \begin{pmatrix} 16 \\ 21 \end{pmatrix},$$

and so $C = AP$ is "QV".

Remark 4.2.7. To encipher a plaintext sequence of $k-$digraphs $P = P_1P_2P_3 \cdots P_k$, we can write the k vectors as columns of a $2\times k$-matrix, which we also denote P, and then multiply the 2×2-matrix A by the $2 \times k$-matrix P to get a $2 \times k$-matrix $C = AP$ of coded digraph-vectors.

Example 4.2.8. Continue as in Example 4.2.5 to encipher the plaintext "NOANSWER."

Solution. The numerical equivalent of "NOANSWER" is the sequence of vectors $\begin{pmatrix} 13 \\ 14 \end{pmatrix} \begin{pmatrix} 0 \\ 13 \end{pmatrix} \begin{pmatrix} 18 \\ 22 \end{pmatrix} \begin{pmatrix} 4 \\ 17 \end{pmatrix}$. We have

$$C = AP = \begin{pmatrix} 2 & 3 \\ 7 & 8 \end{pmatrix} \begin{pmatrix} 13 & 0 & 18 & 4 \\ 14 & 13 & 22 & 17 \end{pmatrix} = \begin{pmatrix} 68 & 39 & 102 & 59 \\ 203 & 104 & 302 & 164 \end{pmatrix}$$

$$= \begin{pmatrix} 16 & 13 & 24 & 7 \\ 21 & 0 & 16 & 8 \end{pmatrix}$$

i.e., the coded message is "QVNAYQHI"

Example 4.2.9. In the situation of Examples 4.2.5 and 4.2.7, decipher the ciphertext "FWMDIQ."

Solution. We have:

$$P = A^{-1}C = \begin{pmatrix} 14 & 11 \\ 17 & 10 \end{pmatrix} \begin{pmatrix} 5 & 12 & 8 \\ 22 & 3 & 16 \end{pmatrix}$$
$$= \begin{pmatrix} 0 & 19 & 2 \\ 19 & 0 & 10 \end{pmatrix} = \text{"ATTACK"}.$$

Remark 4.2.10. As in section 4.1, suppose that we have some limited information from which we want to analyze how to decipher a string of ciphertext. We know that the "enemy" is using digraph-vectors in an N-letter alphabet and a linear enciphering transformation $C = AP$. However, we do not have the enciphering "key" - the matrix A - or the deciphering "key" - the matrix A^{-1}. But suppose we are able to determine two pairs of plaintext and ciphertext digraphs: $C_1 = AP_1$ and $C_2 = AP_2$. Perhaps we learned this information from an analysis of the frequency of occurrence of digraphs in a long string of ciphertext. Or perhaps we know from some outside source that a certain 4-letter plaintext segment corresponds to a certain 4-letter ciphertext. In that case we can proceed as follows to determine A and A^{-1}. We put the two columns P_1 and P_2

together into a 2×2-matrix P, and similarly for the ciphertext columns. We obtain an equation of 2×2-matrices: $C = AP$, in which C and P are known to us, and A is the unknown. We can solve for A by multiplying both sides by P^{-1} :

$$A = APP^{-1} = CP^{-1}.$$

Similarly, from the equation $P = A^{-1}C$ we can solve for A^{-1}:

$$A^{-1} = PC^{-1}.$$

Example 4.2.11. Suppose that we know that our adversary is using a 2×2 enciphering matrix with a 29-letter alphabet, where A-Z have the usual numerical equivalents, blank=26, ?=27, !=28. We receive the message

"GFPYJP X?UYXSTLADPLW,"

and we suppose that we know that the last five letters of plaintext are our adversary's signature "KARLA." Since we don't know the sixth letter from the end of the plaintext, we can only use the last four letters to make two digraphs of plaintext. Thus, the ciphertext digraphs DP and LW correspond to the plaintext digraphs AR and LA, respectively. That is, the matrix P made up from AR and LA is the result of applying the unknown deciphering matrix A^{-1} to the matrix C made up from DP and

LW:

$$\begin{pmatrix} 0 & 11 \\ 17 & 0 \end{pmatrix} = A^{-1} \begin{pmatrix} 3 & 11 \\ 15 & 22 \end{pmatrix}$$

Thus,

$$A^{-1} = \begin{pmatrix} 0 & 11 \\ 17 & 0 \end{pmatrix} \begin{pmatrix} 3 & 11 \\ 15 & 22 \end{pmatrix}^{-1} = \begin{pmatrix} 0 & 11 \\ 17 & 0 \end{pmatrix} \begin{pmatrix} 3 & 13 \\ 23 & 7 \end{pmatrix} = \begin{pmatrix} 21 & 19 \\ 22 & 18 \end{pmatrix},$$

and the full plaintext message is

$$\begin{pmatrix} 21 & 19 \\ 22 & 18 \end{pmatrix} \begin{pmatrix} 6 & 15 & 9 & 26 & 27 & 24 & 18 & 11 & 3 & 11 \\ 5 & 24 & 15 & 23 & 20 & 23 & 19 & 0 & 15 & 22 \end{pmatrix}$$

$$= \begin{pmatrix} 18 & 17 & 10 & 26 & 19 & 13 & 14 & 28 & 0 & 11 \\ 19 & 8 & 4 & 0 & 26 & 14 & 13 & 10 & 17 & 0 \end{pmatrix}$$

$$= \text{"STRIKE AT NOON!KARLA."}$$

Remark 4.2.12. In order for this to work, notice that the matrix P formed by the two known plaintext digraphs must be invertible, i.e., its determinant D must have no common factor with the number of letters N. What if we are not so fortunate? If we happen to know another ciphertext-plaintext pair,hen we could try to use that pair of columns in place of either the first or second columns of P and C, hoping to obtain then an invertible matrix. But suppose we have no further information, or that none of the known plaintext digraphs give us an invertible matrix P. Then we cannot find A^{-1} exactly. However, we might be able to

get enough information about A^{-1} to cut down drastically the number of possibilities for the deciphering matrix. We now illustrate this with an example.

Example 4.2.13. Suppose we know that our adversary is using an enciphering matrix A in the 26-letter alphabet. We intercept the ciphertext "WKNCCHSSJH," and we know that the first word is "GIVE." We want to find the deciphering matrix A^{-1} and read the message.

Solution. If we try to proceed as in Example 4.2.11, writing

$$P = \text{“GIVE”} = \begin{pmatrix} 6 & 21 \\ 8 & 4 \end{pmatrix}, C = \text{”WKNC”} = \begin{pmatrix} 22 & 13 \\ 10 & 2 \end{pmatrix}, \quad \text{and}$$
$$A^{-1} = PC^{-1},$$

we immediately run into a problem, since $det(C) = 18$ and $g.c.d.(18, 26) = 2$. We can proceed as follows. Let $\bar{A}$ denote the reduction modulo 13 of the matrix A, and similarly for $\bar{P}$ and $\bar{C}$. If we consider these matrices in $M_2(\mathbf{Z}/13\mathbf{Z})$, we can take C^{-1} (more precisely, $\bar{C}^{-1}$), because $g.c.d.(det(C), 13) = 1$. Thus, from $\bar{P} = \bar{A}^{-1}\bar{C}$ we can compute

$$\bar{A}^{-1} = \bar{P}\bar{C}^{-1} = \begin{pmatrix} 6 & 8 \\ 8 & 4 \end{pmatrix} \begin{pmatrix} 9 & 0 \\ 10 & 2 \end{pmatrix}^{-1} = \begin{pmatrix} 2 & 4 \\ 3 & 2 \end{pmatrix}$$

Since the entries of A^{-1}, which are integers mod 26, must reduce to

$$\begin{pmatrix} 2 & 4 \\ 3 & 2 \end{pmatrix}$$

modulo 13, it follows that there are two possibilities for each entry in the

matrix A^{-1}. More precisely,

$$A^{-1} = \begin{pmatrix} 2 & 4 \\ 3 & 2 \end{pmatrix} + 13A_1,$$

where $A_1 \in M_2(\mathbf{Z}/2\mathbf{Z})$ is a $2\times$ 2-matrix of 0's and 1's. That leaves $2^4 = 16$ possibilities. However, in the first place, since A^{-1} is invertible, its determinant must be prime to 26, and hence also prime to 2 (i.e., odd). This consideration rules out all but 6 possibilities for A_1. In the second place, when we substitute

$$\begin{pmatrix} 2 & 4 \\ 3 & 2 \end{pmatrix} + 13A_1$$

for A^{-1} in the equation

$$A^{-1} \begin{pmatrix} 22 & 13 \\ 10 & 2 \end{pmatrix} = \begin{pmatrix} 6 & 21 \\ 8 & 4 \end{pmatrix}$$

(this means entry-by-entry congruence mod 26), we eliminate all but 2 possibilities, namely,

$$A_1 = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \qquad \text{or} \qquad \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}, \text{i.e.,}$$

$$A^{-1} = \begin{pmatrix} 15 & 4 \\ 16 & 15 \end{pmatrix} \qquad \text{or} \qquad \begin{pmatrix} 15 & 17 \\ 16 & 15 \end{pmatrix}.$$

Attempting to decipher with the first matrix yields "GIVEGHEMHP,"

which must be wrong. Deciphering with the second matrix

$$A^{-1} = \begin{pmatrix} 15 & 17 \\ 16 & 15 \end{pmatrix}$$

leads to "GIVETHEMUP." So that must be correct. Although a certain amount of trial and error is involved, it's better than running through all 157,248 possibilities for a deciphering matrix $A^{-1} \in M_2(\mathbf{Z}/26\mathbf{Z})^*$.

Remark 4.2.14. In Example 4.2.13 it would perhaps be more efficient to adjust the 1 entries in $\bar{A}^{-1}$ by multiples of 13 so that they become divisible by 2, i.e., to define A_1 by writing:

$$A^{-1} = \begin{pmatrix} 2 & 4 \\ 16 & 2 \end{pmatrix} + 13A_1.$$

Then one can obtain information on A_1 by working modulo 2, since we now have $A_1 C \equiv P \mod 2$.

Affine enciphering transformations. A more general way to encipher a digraph-vector $P = \begin{pmatrix} x \\ y \end{pmatrix}$ is to apply a 2×2-matrix $A = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in M_2(\mathbf{Z}/N\mathbf{Z})$ and then add a constant vector $B = \begin{pmatrix} e \\ f \end{pmatrix}$:

$$C = AP + B$$

i.e.,

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix} = \begin{pmatrix} ax + by + e \\ cx + dy + f \end{pmatrix}.$$

This is called an "affine" map, and is analogous to the enciphering function $C = aP + b$ that we studied in section 4.1 when we were using single-letter message units. Of course, as before, we are using "=" to mean the corresponding entries are congruent mod N.

The inverse transformation that expresses P in terms of C can be found by subtracting B from both sides and then applying A^{-1} to both sides:

$$P = A^{-1}C - A^{-1}B$$

This is also an affine transformation $P = A'C + B'$, where $A' = A^{-1}$ and $B' = -A^{-1}B$. Notice that we must assume that A is an invertible matrix in order to be able to decipher uniquely.

Suppose we know that our adversary is using an fine enciphering transformation of digraph-vectors with an N-letter alphabet. To determine A and B (or to determine $A' = A^{-1}$ and $B' = -A^{-1}B$), we need at least three digraph pairs. Suppose we know that the ciphertext digraphs C_1, C_2, C_3 correspond to the plaintext digraphs P_1, P_2, P_3:

$$P_1 = A'C_1 + B'$$

$$P_2 = A'C_2 + B'$$

$$P_3 = A'C_3 + B'.$$

To find A' and B' we can proceed as follows. Subtract the last equation from the first two, and then make a 2×2-matrix P from the two columns $P_1 - P_3$ and $P_2 - P_3$ and a 2×2-matrix C from the two columns $C_1 - C_3$ and $C_2 - C_3$. We obtain the matrix equation $P = A'C$, which can be solved for A' (provided that C is invertible) as we did in the case of linear enciphering transformations. Finally, once we find $A' = A^{-1}$, we can determine B' from any of the above three equations, e.g., $B' = P_1 - A'C_1$.

Let Us Sum Up

- An enciphering map is a 1-to-l function from $\left(\mathbf{Z}/N\mathbf{Z}\right)^2$ to itself.
- By the inverse of the matrix A we mean the matix which multiplies by it to give the identity matrix $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$.
- $A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ has an inverse if and only if its determinant $D =_{def}$ $ad - bc$ is nonzero.
- $\left(\mathbf{Z}/N\mathbf{Z}\right)$ is always a ring, but it is not a field unless N is prime.
- To encipher a digraph-vector $P = \begin{pmatrix} x \\ y \end{pmatrix}$, apply a 2×2-matrix $A = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in M_2(\mathbf{Z}/N\mathbf{Z})$ and then add a constant vector $B = \begin{pmatrix} e \\ f \end{pmatrix}$ such that C = AP + B

Check your progress 4.2.

1. Find the inverses of the following matrices mod N. Write the entries in the inverse matrix as nonnegative integers less than N.
   (a) $\begin{pmatrix} 1 & 3 \\ 4 & 3 \end{pmatrix}$ mod 5 (b) $\begin{pmatrix} 1 & 3 \\ 4 & 3 \end{pmatrix}$ mod 29
   (c) $\begin{pmatrix} 15 & 17 \\ 4 & 9 \end{pmatrix}$ mod 26 (d) $\begin{pmatrix} 40 & 0 \\ 0 & 21 \end{pmatrix}$ mod 841
   (e) $\begin{pmatrix} 197 & 62 \\ 603 & 271 \end{pmatrix}$ mod 841

Unit Summary

In this unit we have studied about Cryptosystem, Cryptanalysis and Digraph transformation. Also we have learned about Enciphering Matrices, Review of linear algebra, Linear algebra modulo and Affine enciphering transformations.

Glossary

Cryptography - The study of methods of sending messages in disguised form.

Encryption - The process of converting plaintext into ciphertext.

Decryption - The process of converting ciphertext into plaintext.

Affine map - A function that combines a linear transformation with a translation.

Digraph - Two-letter blocks.

Exercise 4.

1. Using frequency analysis, cryptanalyze and decipher the following message, which you know was enciphered using a shift transformation of single-letter plaintext message units in the 26-letter alphabet:

   PXPXKXENVDRUXVTNLXHYMXGMAXYKXJN

   XGVRFXMAHWGXXWLEHGZXKVBIAXKMXQM.

2. In a long string of ciphertext which was encrypted by means of an affine map on single-letter message units in the 26-letter alphabet, you observe that the most frequently occurring letters are "Y" and "V", in that order. Assuming that those ciphertext message units are the encryption of "E" and "T", respectively, read the message "QAOOYQQEVHEQV".

3. You are trying to cryptanalyze an affine enciphering transformation of single-letter message units in a 37-letter alphabet. This alphabet includes the numerals 0-9, which are labeled by themselves (i.e., by the integers 0-9). The letters A-Z have numerical equivalents 10-35, respectively, and blank=36. You intercept the ciphertext "OH7F86BB46R3627O266BB9" (here the O's are the letter "oh", not the numeral zero). You know that the plaintext ends with the signature "007" (zero zero seven). What is the message?

4. You intercept the ciphertext "OFJDFOHFXOL", which was enciphered using an affine transformation of single-letter plaintext units in the 27-letter alphabet (with blank=26). You know that the first word is "I " ("I" followed by blank). Determine the enciphering key, and read the message.

5. A plaintext message unit P is said to be fixed for a given enciphering transformation f if $f(P) = P$. Suppose we are using an affine enciphering transformation on single-letter message units in an N-letter alphabet. In this problem we also assume that the affine map is not a shift, i.e., that $a \neq 1$.

   (a) Prove that if N is a prime number, then there is always exactly one fixed letter.

   (b) Prove (for any N) that if our affine transformation is linear, i.e., if $b = 0$, then it has at least one fixed letter; and that, if N is even, then a linear enciphering transformation has at least two fixed letters.

   (c) Give an example for some N of an affine enciphering transformation which has no fixed letter.

6. Now suppose that our message units are digraphs in an N-letter alphabet. Find a formula for the number of different affine enciphering transformations there are. How many are there when $N = 26, 27, 29, 30$?

7. (a)$x + 4y \equiv 1 \mod 9, \quad 5x + 7y \equiv 1 \mod 9$

(b)$x + 4y \equiv 1 \mod 9 \quad 5x + 8y \equiv 1 \mod 9$

(c) $x + 4y \equiv 1 \mod 9 \quad 5x + 8y \equiv 2 \mod 9$

(d)$x + 4y \equiv 0 \mod 9 \quad 5x + 8y \equiv 0 \mod 9$

8. (a) $17x + 11y \equiv 7 \mod 29 \quad 13x + 10y \equiv 8 \mod 29$

   (b) $17x + 11y \equiv 0 \mod 29 \quad 13x + 10y \equiv 0 \mod 29$

   (c) $9x + 13y \equiv 0 \mod 29 \quad 16x + 13y \equiv 0 \mod 29$

   (d)$9x + 20y \equiv 10 \mod 29 \quad 16x + 13y \equiv 21 \mod 29$

   (e)$9x + 20y \equiv 1 \mod 29 \quad 16x + 13y \equiv 2 \mod 29$

9. (a)$480x + 971y \equiv 416 \mod 1111 \quad 297x + 398y \equiv 319 \mod 1111$

   (b)$480x + 971y \equiv 109 \mod 1111 \quad 297x + 398y \equiv 906 \mod 1111$

   (c)$480x + 971y \equiv 0 \mod 1111 \quad 297x + 398y \equiv 0 \mod 1111$

   (d)$480x + 971y \equiv 0 \mod 1111 \quad 298x + 398y \equiv 0 \mod 1111$

   (e)$480x + 971y \equiv 648 \mod 1111 \quad 298x + 398y \equiv 1004 \mod 1111$

10. The *Fibonnacci* numbers can be defined by the rule $f_1 = 1$, $f_2 = 1$, $f_3 = 2$, $f_{n+1} = f_n + f_{n-1}$ for $n > 1$, or, equivalently, by means of the matrix equation

$$\begin{pmatrix} f_{n+1} & f_n \\ f_n & f_{n-1} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^n .$$

Using matrix form of the definition, prove that f_n is even if and only if n is divisible by 3. More generally, prove that f_n is divisible by a

if and only if n is divisible by b for the following a and b: (a)$a = 2$, $b = 3$; (b)$a = 3$, $b = 4$; (c)$a = 5$, $b = 5$; (d)$a = 7$, $b = 8$; (e) $a = 8$, $b = 6$; (f)$a = 11$, $b = 10$.

11. You intercept the message "SONAFQCHMWPTVEVY" which you know resulted from a linear enciphering transformation of digraph-vectors, where the sender used the usual 26-letter alphabet A-Z with numerical equivalents 0-25, respectively. An earlier statistical analysis of a long string of intercepted ciphertext revealed that the most frequently occurring ciphertext digraphs were "KH" and "XW" in that order. You take a guess that those digraphs correspond to "TH" and "HE," respectively, since those are the most frequently occurring digraphs in most long plaintext messages on the subject you think is being discussed. Find the deciphering matrix, and read the message.

12. You intercept the message "ZRIXXYVBMNPO," which you know resulted from a linear enciphering transformation of digraph-vectors in a 27-letter alphabet, in which A-Z have numerical equivalents 0-25, and blank=26. You have found that the most frequently occurring ciphertext digraphs are "PK" and "RZ." You guess that they correspond to the most frequently occurring plaintext digraphs in the 27-letter alphabet, namely, "E " (E followed by blank) and "S ." Find the deciphering matrix, and read the message.

13. You intercept the message "!IWGVIEX!ZRADRYD," which was sent using a linear enciphering transformation of digraph-vectors in a

29- letter alphabet, in which A-Z have numerical equivalents 0-25, blank=26, ?=27, !=28. You know that the last five letters of plaintext are the sender's signature "MARIA."

(a) Find the deciphering matrix, and read the message.

(b) Find the enciphering matrix, and, impersonating Maria's friend Jo, send the following reply in code: "DAMN FOG! JO."

Answers.

Check your progress 4.1

1. THRPXDH.
2. N.
3. $N\varphi(N) = N^2 \prod_{p|N}(1 - \frac{1}{p})$.
4. 312, 486, 812, 240.

Check your progress 4.2

1. (a) $\begin{pmatrix} 3 & 2 \\ 1 & 1 \end{pmatrix}$; (b) $\begin{pmatrix} 19 & 10 \\ 23 & 16 \end{pmatrix}$; (c) $\begin{pmatrix} 11 & 11 \\ 24 & 1 \end{pmatrix}$; (d) $\begin{pmatrix} 820 & 0 \\ 0 & 801 \end{pmatrix}$; (e) $\begin{pmatrix} 127 & 303 \\ 546 & 353 \end{pmatrix}$.

Exercise 4.

1. Use the fact that "X" occurs most frequently in the ciphertext to find that $b = 19$. The message is: WEWERELUCKYBECAUSEOFTENTHEFREQUENCYMETHODNEEDSLONGERCIPHERTEXT.

2. SUCCESSATLAST.

3. AGENT 006 IS DEAD 007.

4. You find 9 possibilities for a' and b': $a' = 1, 4, 7, 10, 13, 16, 19, 22, 25$, and $b' = 21, 6, 18, 3, 15, 0, 12, 24, 9$, respectively. Since you have no more information to go on, simply try all nine possibilities; it turns out that only the third one $P \equiv 7C + 18 \mod 27$ gives a meaningful plaintext. The plaintexts of the nine tranformations are, respectively: "I DY IB RIF," "I PS IH RIX;" "I AM IN RIO," "I MG IT RIF," "I YA IZ RIX," "I JV IE RIO," "I VP IK RIF," "I GJ IQ RIX," "I SD IW RIO".

5. (a) If $a \neq 1$, then the congruence $(a-l)P \equiv -b \mod N$ has exactly one solution in the field $\mathbf{F}_N = \mathbf{Z}/N\mathbf{Z}$. (b) $P = 0$ is always fixed; for N even (so a must be odd) the congruence $(a-l)P \equiv 0 \mod N$ at least has the two solutions $P = 0$ and $P = N/2$. (c) Any example with N even and b odd; more generally, any example in which b is not divisible by $g.c.d.(a-1, N)$.

6. $N^2\varphi(N^2) = N^4 \prod_{p|N}(1 - \frac{1}{p})$; 210,912; 354,294; 682,892; 216,000.

7. (a) $\begin{pmatrix} 6 \\ 1 \end{pmatrix}$; (b) none (since multiplying the second congruence by 2 and subtracting from the first gives $6y \equiv 8 \mod 9$, which woud mean $3|8$); (c) $\begin{pmatrix} 6 \\ 1 \end{pmatrix}, \begin{pmatrix} 3 \\ 4 \end{pmatrix}, \begin{pmatrix} 0 \\ 7 \end{pmatrix}$; (d) $\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 6 \\ 3 \end{pmatrix}, \begin{pmatrix} 3 \\ 6 \end{pmatrix}$.

8. (a) $\begin{pmatrix} 9 \\ 21 \end{pmatrix}$; (b) $\begin{pmatrix} 0 \\ 0 \end{pmatrix}$; (c) any vector with $y = x$, i.e., $\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 2 \\ 2 \end{pmatrix}$, etc.; (d) any vector of the form $\begin{pmatrix} n \\ 15+n \end{pmatrix}$; (e) none.

9. (a) $\begin{pmatrix} 787 \\ 759 \end{pmatrix}$;(b) $\begin{pmatrix} 626 \\ 233 \end{pmatrix}$; (c) $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$; (d) $\begin{pmatrix} 101 \\ 505 \end{pmatrix}, \begin{pmatrix} 202 \\ 1010 \end{pmatrix}, \begin{pmatrix} 303 \\ 404 \end{pmatrix}$, $\begin{pmatrix} 404 \\ 909 \end{pmatrix}, \begin{pmatrix} 505 \\ 303 \end{pmatrix}, \begin{pmatrix} 606 \\ 808 \end{pmatrix}, \begin{pmatrix} 707 \\ 202 \end{pmatrix}, \begin{pmatrix} 808 \\ 707 \end{pmatrix}, \begin{pmatrix} 909 \\ 101 \end{pmatrix}, \begin{pmatrix} 1010 \\ 606 \end{pmatrix}$; (e) add $\begin{pmatrix} 31 \\ 800 \end{pmatrix}$ to any of the 11 vectors of part (d) and reduce mod 1111.

10. Use mathematical induction, proving the assertion for $n = 1, 2, \cdots, b$ by inspection and then proving that the assertion for n implies the assertion for $n + b$. Namely, compute:

$$\begin{aligned}
\begin{pmatrix} f_{n+b+1} & f_{n+b} \\ f_{n+b} & f_{n+b-1} \end{pmatrix} &= \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^{n+b} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^{b} \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^{n} \\
&= \begin{pmatrix} f_{b+1} & f_b \\ f_b & f_{b-1} \end{pmatrix} \begin{pmatrix} f_{n+1} & f_n \\ f_n & f_{n-1} \end{pmatrix} \\
&\equiv \begin{pmatrix} c & 0 \\ 0 & c \end{pmatrix} \begin{pmatrix} f_{n+1} & f_n \\ f_n & f_{n-1} \end{pmatrix} \\
&= \begin{pmatrix} cf_{n+1} & cf_n \\ cf_n & cf_{n-1} \end{pmatrix} \mod a,
\end{aligned}$$

where $c \in (\mathbf{Z}/a\mathbf{Z})^*$, and use the induction assumption. (It can be proved that for any integer a there is an integer b such that $a|f_n \iff b|n$, and that if $a = p^\alpha$ is a power of a prime $p \neq 5$, then b is a divisor of $p^{\alpha-1}(p^2-1)$; the proof uses a little algebraic number theory in the real quadratic field generated by the golden ratio - note that the golden ratio and its conjugate are the eigenvalues of the matrix in the definition of Fibonacci numbers.)

11. $A^{-1} = \begin{pmatrix} 23 & 7 \\ 18 & 5 \end{pmatrix}$, "SENATORTOOK."

12. $A^{-1} = \begin{pmatrix} 22 & 16 \\ 21 & 17 \end{pmatrix}$, "MEET AT NOON."

13. $A^{-1} = \begin{pmatrix} 22 & 20 \\ 28 & 8 \end{pmatrix}$, "WHY NO GO? MARIA"; $A = \begin{pmatrix} 3 & 7 \\ 4 & 1 \end{pmatrix}$, "JMLD W EFWJV."

References:

1. Neal Koblit, A course in Number Theory and Cryptography, Springer - Verlag, New York, 2nd edition, 2002.

Suggested Reading:

1. I. Niven and H. S. Zuckermann, An Introduction to Theory of Numbers (Edition 3), Wiley Eastern Ltd, New Delhi 1976

2. D. M. Burton, Elementary Number Theory, Brown Publishers, Iowa, 1989

3. K. Ireland and M. Rosen, A classic Introduction to Modern Number Theory, Springer - Verlag, 1972

4. N. Koblit, Algebraic Aspects of Cryptography, Springer-Verlag, 1998.

UNIT - 5

Unit 5

Public Key Cryptography

Objectives.

By studying this unit, the students will

1. recall the cryptosystem.
2. understand the idea of Public key and cryptography.
3. know trapdoor function.
4. understand the hash functions.
5. know the key exchange and Probabilistic Encryption.
6. learn RSA cryptosystem.

5.1 The idea of public key cryptography.

Recall that a cryptosystem consists of a 1-to-1 enciphering transformation f from a set $\mathcal{P}$ of all possible plaintext message units to a set $\mathcal{C}$ of all possible ciphertext message units. Actually, the term "cryptosystem"

is more often used to refer to a whole family of such transformations, each corresponding to a choice of *parameters* (the sets $\mathcal{P}$ and $\mathcal{C}$, as well as the map f, may depend upon the values of the parameters). For example, for a fixed N-letter alphabet (witn numerical equivalents also fixed once and for all), we might consider the affine cryptosystem (or "family of cryptosystems") which for each $a \in \left(\mathbf{Z}/N\mathbf{Z}\right)^*$ and $b \in \mathbf{Z}/N\mathbf{Z}$ is the map from $\mathcal{P} = \mathbf{Z}/N\mathbf{Z}$ to $\mathcal{C} = \mathbf{Z}/N\mathbf{Z}$ defined by $C \equiv aP + b \mod N$. In this example, the sets $\mathcal{P}$ and $\mathcal{C}$ are fixed (because N is fixed), but the enciphering transformation f depends upon the choice of parameters a, b. The enciphering transformation can then be described by (i) an algorithm, which is the same for the whole family, and (ii) the values of the parameters. The values of the parameters are called the *enciphering key* K_E. In our example, K_E is the pair (a, b). In practice, we shall suppose that the algorithm is publicly known, i.e., the general procedure used to encipher cannot he kept secret. However, the keys can easily be changed periodically and, if one wants, kept secret.

One also needs an algorithm and a key in order to decipher, i.e., compute f^{-1}. The key is called the *deciphering key* K_D. In our example of the affine cryptosystem family, deciphering is also accomplished by an affine map, namely $P \equiv a^{-1}C - a^{-1}b \mod N$, and so the deciphering transformation uses the same algorithm as the enciphering transformation except with a different key, namely, the pair $(a^{-1}, -a^{-1}b)$. (In some cryptosystems, the deciphering algorithm, as well as the key, is different from the enciphering algorithm.) We shall always suppose that the deci-

phering and enciphering algorithms are publicly known, and that it is the keys K_E and K_D which can be concealed.

Let us suppose that someone wishes to communicate secretly using the above affine cryptosystem $C \equiv aP+b$. We saw in section 4.1 that it is not hard to break the system if one uses single-letter message units in an N-letter alphabet. It is a little more difficult to break the system if one uses digraphs, which can be regarded as symbols in an N^2-letter alphabet. It would be safer to use blocks of k letters, which have numerical equivalents in $\mathbf{Z}/N^k\mathbf{Z}$. At least for $k > 3$ it is not easy to use frequency analysis, since the number of possible k-letter blocks is very large, and one will find many that are close contenders for the title of most frequently occurring k-graph. If we want to increase k, we must be concerned about the length of time it takes to do various arithmetic tasks (the most important one being finding a^{-1} by the Euclidean algorithm) involved in setting up our keys and carrying out the necessary transformations every time we send a message or our friend at the other end deciphers a message from us. That is, it is useful to have big-O estimates for the order of magnitude of time (as the parameters increase, i.e., as the cryptosystem becomes "larger") that it takes to: encipher (knowing K_E), decipher (knowing K_D), or break the code by enciphering without knowledge of K_E or deciphering without knowledge of K_D.

In all of the examples in Uint IV - and in all of the cryptosystems used historically until about fifteen years ago - it is not really necessary to specify the deciphering key once the enciphering key (and the general

algorithms) are known. Even if we are working with large numbers - such as N^k with k fairly large - it is possible to determine the deciphering key from the enciphering key using an order of magnitude of time which is roughly the same as that needed to implement the various algorithms. For example, in the case of an affine enciphering transformation of $\mathbf{Z}/N^k\mathbf{Z}$, once we know the enciphering key $K_E = (a, b)$ we can compute the deciphering key $K_D = (a^{-1} \mod N^k, -a^{-1}b \mod N^k)$ by the Euclidean algorithm in $O(log^3(N^k))$ bit operations.

Thus, with a traditional cryptosystem anyone who knew enough to decipher messages could, with little or no extra effort, determine the enciphering key. Indeed, it was considered naive or foolish to think that someone who had broken a cipher might nevertheless not know the enciphering key. We see this in the following passage from the autobiography of a well-known historical personality: Five or six weeks later, she [Madame d′Urfé] asked me if I had deciphered the manuscript which had the transmutation procedure. I told her that I had.

"Without the key, sir, excuse me if I believe the thing impossible."

"Do you wish me to name your key, madame?"

"If you please."

I then told her the key-word, which belonged to no language, and I saw her surprise. She told me that it was impossible, for she believed herself the only possessor of that word which she kept in her memory and which she had never written down.

I could have told her the truth - that the same calculation which had

served me for deciphering the manuscript had enabled me to learn the word - but on a caprice it struck me to tell her that a genie had revealed it to me. This false disclosure fettered Madame d'Urfé to me. That day I became the master of her soul, and I abused my power. Every time I think of it, I am distressed and ashamed, and I do penance now in the obligation under which I place myself of telling the truth in writing my memoirs.

- Casanova, 1757, quoted in D. Kahn's *The Codebreakers*

The situation persisted for another 220 years after this encounter between Casanova and Madame d′Urfé: knowledge of how to encipher and knowledge of how to decipher were regarded as essentially equivalent in any cryptosystem. However, in 1976 W. Diffie and M. Hellman discovered an entirely different type of cryptosystem and invented "public key cryptography."

By definition, a **public key cryptosystem** has the property that someone who knows only how to encipher cannot use the enciphering key to find the deciphering key without a prohibitively lengthy computation. In other words the enciphering function $f : \mathcal{P} \longrightarrow \mathcal{C}$ is easy to compute once the enciphering key K_E is known, but it is very hard in practice to compute the inverse function $f^{-1} : \mathcal{C} \longrightarrow \mathcal{P}$. That is, from the standpoint of realistic computability, the function f is not invertible (without some additional information - the deciphering key K_D). Such a function f is called a **trapdoor function**. That is, a trapdoor function f is a function which is easy to compute but whose inverse f^{-1} is hard to com-

pute without having some additional auxiliary information beyond what is necessary to compute f. The inverse f^{-1} is easy to compute, however, for someone who has this information K_D (the "deciphering key").

There is a closely related concept of a one-way function. This is a function f which is easy to compute but for which f^{-1} is hard to compute and cannot be made easy to compute even by acquiring some additional information. While the notion of a trapdoor function apparently appeared for the first time in 1978 along with the invention of the RSA public-key cryptosystem, the notion of a one-way function is somewhat older. What seems to have been the first use of one-way functions for cryptography was described in Wilkes' book about time-sharing systems that was published in 1968. The author describes a new **one-way cipher** used by R. M. Needham in order to make it possible for a computer to verify passwords without storing information that could be used by an intruder to impersonate a legitimate user. In Needham's system, when the user first sets his password, or whenever he changes it, it is immediately subjected to the enciphering process, and it is the enciphered form that is stored in the computer. Whenever the password is typed in response to a demand from the supervisor for the user's identity to be established, it is again enciphered and the result compared with the stored version. It would be of no immediate use to a would-be malefactor to obtain a copy of the list of enciphered passwords, since he would have to decipher them before he could use them. For this purpose, he would need access to a computer and even if full details of the enciphering algorithm were available, the

deciphering process would take a long time.

In 1974, G. Purdy published the first detailed description of such a one-way function. The original passwords and their enciphered forms are regarded as integers modulo a large prime p, and the "one-way" map $\mathbf{F}_p \longrightarrow \mathbf{F}_p$ is given by a polynomial $f(x)$ which is not hard to evaluate by computer but which takes an unreasonably long time to invert. Purdy used $p = 2^{64} - 59$, $f(x) = x^{2^{24}+17} + a_1 x^{2^{24}+3} + a_2 x^3 + a_3 x^2 + a_4 x + a_5$, where the coefficients a_i were arbitrary 19-digit integers.

The above definitions of a public key cryptosystem and a one-way or trapdoor function are not precise from a rigorous mathematical standpoint. The notion of "realistic computability" plays a basic role. But that is an empirical concept that is affected by advances in computer technology (e.g., parallel processor techniques) and the discovery of new algorithms which speed up the performance of arithmetic tasks (sometimes by a large factor). Thus, it is possible that an enciphering transformation that can safely be regarded as a one-way or trapdoor function in 1994 might lose its one-way or trapdoor status in 2004 or in the year 2994.

It is conceivable that some transformation could be proved to be trapdoor. That is, there could be a theorem that provides a nontrivial lower bound for the number of bit operations that would be required ("on the average," i.e., for random values of the key parameters) in order to figure out and implement a deciphering algorithm without the deciphering key. Here one would have to allow the possibility of examining a large number

of corresponding plaintext-ciphertext message units (as in our frequency analysis of the simple systems in Unit IV), because, by the definition of a public key system, any user can generate an arbitrary number of plaintext-ciphertext pairs. One would also have to allow the use of "probabilistic" methods which, while not guaranteed to break the code at once, would be likely to work if repeated many times. Unfortunately, no such theorems have been proved for any of the functions that have been used as enciphering maps. Thus, while there are now many cryptosystems which empirically seem to earn the right to be called "public key," there is no cryptosystem in existence which is *provably* public key.

The reason for the name "public key" is that the information needed to send secret messages - the enciphering key K_E - can be made public information without enabling anyone to read the secret messages. That is, suppose we have some population of users of the cryptosystem, each one of whom wants to be able to receive confidential communications from any of the other users without a third party (either another user or an outsider) being able to decipher the message. Some central office can collect the enciphering key $K_{E,A}$ from each user A and publish all of the keys in a "telephone book" having the form

AAA Banking Company	(9974398087453939, 2975290017591012)
Aardvark, Aaron	(8870004228331, 7234752637937)
⋮	⋮

Someone wanting to send a message merely has to look up the enciphering key in this "telephone book" and then use the general enciphering

algorithm with the key parameters corresponding to the intended recipient. Only the intended recipient has the matching deciphering key needed to read the message.

In earlier ages this type of system would not have seemed to have any particularly striking advantages. Traditionally, cryptography was used mainly for military and diplomatic purposes. Usually there was a small, well-defined group of users who could all share a system of keys, and new keys could be distributed periodically (using couriers) so as to keep the enemy guessing.

However, in recent years the actual and potential applications of cryptography have expanded to include many other areas where communication systems play a vital role - collecting and keeping records of confidential data, electronic financial transactions, and so on. Often one has a large network of users, any two of whom should be able to keep their communications secret from all other users as well as intruders from outside the network. Two parties may share a secret communication on one occasion, and then a little later one of them may want to send a confidential message to a third party. That is, the "alliances" - who is sharing a secret with whom - may be continually shifting. It might be impractical always to be exchanging keys with all possible confidential correspondents.

Notice that with a public key system it is possible for two parties to initiate secret communications without ever having had any prior contact, without having established any prior trust for one another, without exchanging any preliminary information. All of the information necessary

to send an enciphered message is publicly available.

Classical versus public key. By a **classical cryptosystem** (also called a **private key cryptosystem** or a **symmetrical cryptosystem**), we mean a cryptosystem in which, once the enciphering information is known, the deciphering transformation can be implemented in approximately the same order of magnitude of time as the enciphering transformation. All of the cryptosystems in Unit IV are classical. Occasionally, it takes a little longer for the deciphering - because one needs to apply the Euclidean algorithm to find an inverse modulo N or one must invert a matrix (and this can take a fairly long time if we work with $k \times k$ -matrices for k larger than 2) - nevertheless, the additional time required is not prohibitive. (Moreover, usually the additional time is required only once - to find K_D- after which it takes no longer to decipher than to encipher.) For example, we might need only $O(log^2 B)$ to encipher a message unit, and $O(log^3 B)$ bit operations to decipher one by finding K_D from K_E, where B is a bound on the size of the key parameters. Notice the role of big-O estimates here.

If, on the other hand, the enciphering time were polynomial in $\log B$ and the deciphering time (based on knowledge of K_E but not K_D) were, say, polynomial in B but not in $\log B$, then we would have a *public key* rather than a classical cryptosystem.

Authentication. Often, one of the most important parts of a message is the **signature**. A person's signature - hopefully, written with an idiosyn-

cratic flourish of the pen which is hard to duplicate - lets the recipient know that the message really is from the person whose name is typed below. If the message is particularly important, it might be necessary to use additional methods to *authenticate* the communication. And in electronic communication, where one does not have a physical signature, one has to rely entirely on other methods. For example, when an officer of a corporation wants to withdraw money from the corporate account by telephone, he/she is often asked to give some personal information (e.g., mother's maiden name) which the corporate officer knows and the bank knows (from data submitted when the account was opened) but which an imposter would not be likely to know.

In public key cryptography there is an especially easy way to identify oneself in such a way that no one could be simply pretending to be you. Let A (Alice) and B (Bob) be two users of the system. Let f_A be the enciphering transformation with which any user of the system sends a message to Alice, and let f_B be the same for Bob. For simplicity, we shall assume that the set $\mathcal{P}$ of all possible plaintext message units and the set $\mathcal{C}$ of all possible ciphertext message units are equal, and are the same for all users. Let P be Alice's "signature" (perhaps including an identification number, a statement of the time the message was sent, etc.). It would not be enough for Alice to send Bob the encoded message $f_B(P)$, since everyone knows how to do that, so there would be no way of knowing that the signature was not forged. Rather, at the beginning (or end) of the message Alice transmits $f_B f_A^{-1}(P)$. Then, when Bob deciphers the whole

message, including this part, by applying f_B^{-1}, he finds that everything has become plaintext except for a small section of jibberish, which is $f_A^{-1}(P)$. Since Bob knows that the message is claimed to be from Alice, he applies f_A (which he knows, since Alice's enciphering key is public), and obtains P. Since no one other than Alice could have applied the function f_A^{-1} which is inverted by f_A , he knows that the message was from Alice.

Hash functions. A common way to sign a document is with the help of a **hash function**. Roughly speaking, a hash function is an easily computable map $f : x \mapsto h$ from a very long input x to a much shorter output h (for example, from strings of about 10^6 bits to strings of 150 or 200 bits) that has the following property: *it is not computationally feasible to find two different inputs x and x' such that $f(x') = f(x)$.* If part of Alice's "signature" consists of the hash value $h = f(x)$, where x is the entire text of her message, then Bob can verify not only that the message was really sent by Alice, but also that it wasn't tampered with during transmission. Namely, Bob applies the hash function f to his deciphered plaintext from Alice, and checks that the result agrees with the value h in Alice's signature. By assumption, no tamperer would have been able to change x without changing the value $h = f(x)$.

Key exchange. In practice, the public key cryptosystems for sending messages tend to be slower to implement than the clasical systems that are in current use. The number of plaintext message units per second

that can be transmitted is less. However, even if a network of users feels attached to the traditional type of cryptosystem, they may want to use a public key cryptosystem in an auxiliary capacity to send one another their keys $K = (K_E, K_D)$ for the classical system. Thus, the ground rules for the classical cryptosystem can be agreed upon, and keys can be periodically exchanged, using the slower public key cryptography; while the large volume of messages would then be sent by the faster, older methods.

Probabilistic Encryption. Most of the number theory based cryptosystems for message transmission are *deterministic*, in the sense that a given plaintext will always be encrypted into the same ciphertext any time it is sent. However, deterministic encryption has two disadvantages: (1) if an eavesdropper knows that the plaintext message belongs to a small set (for example, the message is either "yes" or "no"), then she can simply encrypt all possibilities in order to determine which is the supposedly secret message; and (2) it seems to be very difficult to *prove* anything about the security of a system if the encryption is deterministic. For these reasons, *probabilistic encryption* was introduced.

Let Us Sum Up

- Cryptosystem consists of a 1-to-1 enciphering transformation f from a set $\mathcal{P}$ of all possible plaintext message units to a set $\mathcal{C}$ of all possible ciphertext message units.

- The enciphering transformation can be described by (i) an algorithm, which is the same for the whole family, and (ii) the values of the parameters. The values of the parameters are called the enciphering key K_E.

- The enciphering function $f : \mathcal{P} \longrightarrow \mathcal{C}$ is easy to compute once the enciphering key K_E is known, but it is very hard in practice to compute the inverse function $f^{-1} : \mathcal{C} \longrightarrow \mathcal{P}$.

- Classical cryptosystem is also called a private key cryptosystem or a symmetrical cryptosystem.

- In classical cryptosystem if the enciphering information is known, the deciphering transformation can be implemented in approximately the same order of magnitude of time as the enciphering transformation.

- A Hash function is not computationally feasible to find two different inputs x and x' such that $f(x') = f(x)$.

Check your progress 5.1

1. How will you describe the enciphering transformation?

2. What is trapdoor function?

3. What is meant by a classical cryptosystem?

4. What is a hash function?

5. What are the disadvantages of deterministic encryption?

5.2 RSA

In looking for a trapdoor function f to use for a public key cryptosystem, one wants to use an idea which is fairly simple conceptually and lends itself to easy implementation. On the other hand, one wants to have very strong empirical evidence - based on a long history of attempts to find algorithms for $f^{-1}-$ that decryption cannot feasibly be accomplished without knowledge of the secret deciphering key. For this reason it is natural to look at an ancient problem of number theory: the problem of finding the complete factorization of a large composite integer whose prime factors are not known in advance. The success of the so-called **"RSA" cryptosystem** (from the last names of the inventors Rivest, Shamir, and Adleman), which is one of the oldest (16 years old) and most popular public key cryptosystems, is based on the tremendous difficulty of factoring.

We now describe how RSA works. Each user first chooses two extremely large prime numbers p and q (say, of about 100 decimal digits each), and sets $n = pq$. Knowing the factorization of n, it is easy to compute $\varphi(n) = (p-l)(q-1) = n+1-p-q$. Next, the user randomly chooses an integer e between 1 and $\varphi(n)$ which is prime to $\varphi(n)$.

Remark 5.2.1. Whenever we say "random" we mean that the number was chosen with the help of a random-number generator (or "pseudo-random" number generator), i.e., a computer program that generates a

sequence of digits in a way that no one could duplicate or predict, and which is likely to have all of the statistical properties of a truly random sequence. A lot has been written concerning efficient and secure ways to generate random numbers, but we shall not concern ourselves with this question here. In the RSA cryptosystem we need a random number generator not only to choose e, but also to choose the large primes p and q (so that no one could guess our choices by looking at tables of special types of primes, for example, Mersenne primes or factors of $b^k \pm 1$ for small b and relatively small k). What does a "randomly generated" prime number mean? Well, first generate a large random integer m. If m is even, replace m by $m+1$. Then apply suitable *primality tests* to see if the odd number m is prime. If m is not prime, try $m+2$, then $m+4$, and so on, until you reach the first prime number $\geq m$, which is what you take as your "random" prime. According to the Prime Number Theorem, the frequency of primes among the numbers near m is about $1/log(m)$, so you can expect to test $O(logm)$ numbers for primality before reaching the first prime $\geq m$.

Similarly, the "random" number e prime to $\varphi(n)$ can be chosen by first generating a random (odd) integer with an appropriate number of bits, and then successively incrementing it until one finds an e with $g.c.d.(e, \varphi(n)) = 1$. (Alternately, one can perform primality tests until one finds a prime e, say between $max(p, q)$ and $\varphi(n)$; such a prime must necessarily satisfy $g.c.d.(e, \varphi(n)) = 1$.)

Thus, each user A chooses two primes p_A and q_A and a random number

e_A which has no common factor with $(p_A - l)(q_A - 1)$. Next, A computes $n_A = p_A q_A$, $\varphi(n_A) = n_A + 1 - p_A - q_A$, and also the multiplicative inverse of e_A modulo $\varphi(n_A)$: $d_A =_{def} e_A^{-1} \mod \varphi(n_A)$. She makes public the enciphering key $K_{E,A} = (n_A, e_A)$ and conceals the deciphering key $K_{D,A} = (n_A, d_A)$. The enciphering transformation is the map from $\mathbf{Z}/n_A\mathbf{Z}$ to itself given by $f(P) \equiv P^{e_A} \mod n_A$. The deciphering transformation is the map from $\mathbf{Z}/n_A\mathbf{Z}$ to itself given by $f^{-1}(C) \equiv C^{d_A} \mod n_A$. It is not hard to see that these two maps are inverse to one another, because of our choice of d_A. Namely, performing f followed by f^{-1} or f^{-1} followed by f means raising to the $d_A e_A$-th power. But, because $d_A e_A$ leaves a remainder of 1 when divided by $\varphi(n_A)$, this is the same as raising to the 1-st power (see the Corollary 2.1.16 which gives this in the case when P has no common factor with n_A; if $g.c.d.(P, n_A) > 1$).

From the description in the last paragraph, it seems that we are working with sets $\mathcal{P} = \mathcal{C}$ of plaintext and ciphertext message units that vary from one user to another. In practice, we would probably want to choose $\mathcal{P}$ and $\mathcal{C}$ uniformly throughout the system. For example, suppose we are working in an N-letter alphabet. Then let $k < \ell$ be suitably chosen positive integers, such that, for example, N^k and N^ℓ have approximately 200 decimal digits. We take as our plaintext message units all blocks of k letters, which we regard as k-digit base-N integers, i.e., we assign them numerical equivalents between 0 and N^k. We similarly take ciphertext message units to be blocks of ℓ letters in our N-letter alphabet. Then each user must choose his/her large primes p_A and q_A so that $n_A = p_A q_A$

satisfies $N^k < n_A < N^\ell$. Then any plaintext message unit, i.e., integer less than N^k, corresponds to an element in $\mathbf{Z}/n_A\mathbf{Z}$ (for ariy user's n_A); and, since $n_A < N^\ell$, the image $f(P) \in \mathbf{Z}/n_A\mathbf{Z}$ can be uniquely written as an ℓ-letter block. (Not all ℓ-letter blocks can arise - only those corresponding to integers less than n_A for the particular user's n_A.)

Example 5.2.2. Choose $N = 26$, $k = 3$, $\ell = 4$. That is, the plaintext consists of trigraphs and the ciphertext consists of four-graphs in the usual 26-letter alphabet. To send the message "YES" to a user A with enciphering key $(n_A, e_A) = (46927, 39423)$, we first find the numerical equivalent of "YES," namely: $24 \cdot 26^2 + 4 \cdot 26 + 18 = 16346$, and then compute $16346^{39423} \mod 46927$, which is $21166 = 1{\cdot}26^3{+}5{\cdot}26^2{+}8{\cdot}26{+}2 =$ "BFIC."

The recipient A knows the deciphering key $(n_A, d_A) = (46927, 26767)$, and so computes $21166^{26767} \mod 46927 = 16346 =$"YES." How did user A generate her keys? First, she multiplied the primes $p_A = 281$ and $q_A = 167$ to get n_A; then she chose e_A at random (but subject to the condition that $g.c.d.(e_A, 280) = g.c.d.(e_A, 166) = 1$). Then she found $d_A = e_A^{-1} \mod 280 \cdot 166$. The numbers p_A, q_A, d_A remain secret.

Here, the most time-consuming step is modular exponentiation, e.g.,$16346^{39423} \mod 46927$. But this can be done by the repeated squaring method (see section 2.1) in $O(k^3)$ bit operations, where k is the number of bits in our integers. Actually, if we were working with much larger

integers, potentially the most time-consuming step would be for each user A to find two very large primes p_A and q_A. In order to quickly choose suitable very large primes, one must use an efficient primality test.

Remark 5.2.3. In choosing p and q, user A should take care to see that certain conditions hold. The most important are: that the two primes not be too close together (for example, one should be a few decimal digits longer than the other); and that $p-1$ and $q-1$ have a fairly small $g.c.d.$ and both have at least one large prime factor. Some of the reasons for these conditions are indicated in the exercises below. Of course, if someone discovers a factorization method that works quickly under certain other conditions on p and q, then future users of RSA would have to take care to avoid those conditions as well.

Remark 5.2.4. In section 2.1 we saw that, when n is a product of two primes p and q, knowledge of $\varphi(n)$ is equivalent to knowledge of the factorization. Let's suppose now that we manage to break an RSA system by determining a positive integer d such that $a^{ed} \equiv a \mod n$ for all a prime to n. This is equivalent to $ed-1$ being a multiple of the least common multiple of $p-1$ and $q-1$. Knowing this integer $m = ed-1$ is weaker than actually knowing $\varphi(n)$. But we now give a procedure that with a high probability is nevertheless able to use the integer m to factor n.

So suppose we know n – which is a product of two unknown primes - and also an integer m such that $a^m \equiv 1 \mod n$ for all a prime to n. Notice

that any such m must be even (as we see by taking $a = -1$). We first check whether $m/2$ has the same property, in which case we can replace m by $m/2$. If $a^{m/2}$ is not $\equiv 1 \mod n$ for all a prime to n, then we must have $a^{m/2} \not\equiv 1 \mod n$ for at least 50% of the $a's$ in $(\mathbf{Z}/n\mathbf{Z})^*$. Thus, if we test several dozen randomly chosen $a's$ and find that in all cases $a^{m/2} \equiv 1$ mod n, then with very high probability we have this congruence for all a prime to n, and so may replace m by $m/2$. We keep on doing this until we no longer have the congruence when we take half of the exponent. There are now two possibilities:

(i) $m/2$ is a multiple of one of the two numbers $p-1$, $q-1$ (say, $p-1$) but not both. In this case $a^{m/2}$ is always $\equiv 1 \mod p$ but exactly 50% of the time is congruent to -1 rather than $+1$ modulo q.

(ii) $m/2$ is not a multiple of either $p-1$ or $q-1$. In this case $a^{m/2}$ is $\equiv$ 1 modulo both p and q (and hence modulo n) exactly 25% of the time, it is $\equiv -1$ modulo both p and q exactly 25% of the time, and for the remaining 50% of the values of a it is $\equiv 1$ modulo one of the primes and $\equiv -1$ modulo the other prime.

Thus, by trying $a's$ at random with high probability we will soon find an a for which $a^{m/2} - 1$ is divisible by one of the two primes (say, p) but not the other. (Each randomly selected a has a 50% chance of satisfying this statement.) Once we find such an a we can immediately factor n, because $g.c.d.(n, a^{m/2} - 1) = p$.

The above procedure is an example of a *probabilistic algorithm.*

Remark 5.2.5. How do we send a signature in RSA? When discussing authentication in the last section, we assumed for simplicity that $\mathcal{P} = \mathcal{C}$. We have a slightly more complicated set-up in RSA. Here is one way to avoid the problem of different $n'_A s$ and different blocks sizes (k, the number of letters in a plaintext message unit, being less than ℓ, the number of letters in a ciphertext message unit). Suppose that, as in the last section, Alice is sending her signature (some plaintext P) to Bob. She knows Bob's enciphering key $K_{E,B} = (n_B, e_B)$ and her own deciphering key $K_{D,A} = (n_A, d_A)$. What she does is send $f_B f_A^{-1}(P)$ if $n_A < n_B$, or else $f_A^{-1} f_B(P)$ if $n_A > n_B$. That is, in the former case she takes the least positive residue of P^{d_A} modulo n_A; then, regarding that number modulo n_B, she computes $(P^{d_A} \mod n_A)^{e_B} \mod n_B$, which sends as a ciphertext message unit. In this case $n_A > n_B$, she first computes $P^{e_B} \mod n_B$ and then, working modulo n_A, she raises this to the d_A-th power. Clearly, Bob can verify the authenticity of the message in the first case by raising to the d_B-th power modulo n_B and then to the e_A-th power modulo n_A; in the second case he does these two operations in the reverse order.

Let Us Sum Up

- "RSA" cryptosystem is one of the oldest and most popular public key cryptosystems, based on the tremendous difficulty of factoring.
- In the RSA cryptosystem we need a random number generator not only to choose e, but also to choose the large primes p and q.

- The enciphering transformation is the map from $\mathbf{Z}/n_A\mathbf{Z}$ to itself given by $f(P) \equiv P^{e_A} \mod n_A$. The deciphering transformation is the map from $\mathbf{Z}/n_A\mathbf{Z}$ to itself given by $f^{-1}(C) \equiv C^{d_A} \mod n_A$.

Check your progress 5.2

1. How RSA works?

2. What is random-number generator?

Unit Summaary

In this unit we have discussed about the idea of public key cryptography. Also we have studied about Trapdoor function, Classical versus public key, Authentication, Hash functions, Key exchange and Probabilistic Encryption.

Glossary

Trapdoor function - A function that is easy to compute forward but hard to reverse.

One-way cipher - A cryptographic algorithm that transforms input data into hash.

Exercise 5.

1. Suppose that m users want to be able to communicate with one another using a classical cryptosystem. Each user insists on being able

to communicate with each other user without the remaining $m-2$ users eavesdropping. How many keys $K = (K_E, K_D)$ must be developed? How many keys are needed if they are using a public key cryptosystem? How many keys are needed for each type of cryptosystem if $m = lOOO$?

2. Suppose that the following 40-letters is used for all plaintexts and ciphertexts: A -Z with numerical equivalents 0-25, blank=26, .=27, ?=28, $=29, the numerals 0-9 with numerical equivalents 30-39. Suppose that plaintext message units are digraphs and ciphertext message units are trigraphs (i.e., $k = 2$, $\ell = 3$, $40^2 < n_A < 40^3$ for all n_A).

   (a) Send the message "SEND $7500" to a user whose enciphering key is $(n_A, e_A) = (2047, 179)$.

   (b) Break the code by factoring n_A and then computing the deciphering key n_A, d_A).

   (c) Explain why, even without factoring n_A, a codebreaker could find the deciphering key rather quickly. In other words, why (in addition to its small size) is 2047 a particularly bad choice for n_A?

3. Try to break the code whose enciphering key is

   $(n_A, e_A) = (536813567, 3602561)$. Use a computer to factor n_A by the stupidest known algorithm, i.e., dividing by all odd numbers 3, 5, 7,$\cdots$. If you don't have a computer available, try to guess a prime factor of n_A by trying special classes of prime numbers. After

factoring n_A, find the deciphering key. Then decipher the message BNBPPKZAVQZLBJ, under the assumption that the plaintext consists of 6-letter blocks in the usual 26-letter alphabet (converted to an integer between 0 and $26^6 - 1$ in the usual way) and the ciphertext consists of 7-letter blocks in the same alphabet. It should be clear from this exercise that even a 29-bit choice of n_A is far too small.

4. Suppose that both plaintexts and ciphertexts consist of trigraph message units, but while plaintexts are written in the 27-letter alphabet (consisting of A-Z and blank=26), ciphertexts are written in the 28-letter alphabet obtained by adding the symbol "/" (with numerical equivalent 27) to the 27-letter alphabet. We require that each user A choose n_A between $27^3 = 19683$ and $28^3 = 21952$, so that a plaintext trigraph in the 27-letter alphabet corresponds to a residue P modulo n_A, and then $C = P^{e_A} \mod n_A$ corresponds to a ciphertext trigraph in the 28-letter alphabet.

   (a) If your deciphering key is $K_D = (n, d) = (21583, 20787)$, decipher the message "YSNAUOZHXXH " (one blank at the end).

   (b) If in part (a) you know that $\varphi(n) = 21280$, find (i)$e = d^{-1} \mod \varphi(n)$, and (ii) the factorization of n.

5. Let n be any squarefree integer (i.e., product of distinct primes). Let d and e be positive integers such that $de - 1$ is divisible by $p - 1$ for every prime divisor p of n. (For example, this is the case if $de \equiv 1 \mod \varphi(n)$.) Prove that $a^{de} \equiv a \mod n$ for any integer a (whether

or not it has a common factor with n).

6. Prove the statements in Remark 5.2.4 about the percent of the time the different congruences for $a^{m/2}$ occur in cases (i) and (ii).

Answers :

Check your progress 5.1

1. The enciphering transformation can then be described by (i) an algorithm, which is the same for the whole family, and (ii) the values of the parameters.

2. The enciphering function $f : \mathcal{P} \longrightarrow \mathcal{C}$ is easy to compute once the enciphering key K_E is known, but it is very hard in practice to compute the inverse function $f^{-1} : \mathcal{C} \longrightarrow \mathcal{P}$. That is, from the standpoint of realistic computability, the function f is not invertible (without some additional information - the deciphering key K_D). Such a function f is called a **trapdoor function**. That is, a trapdoor function f is a function which is easy to compute but whose inverse f^{-1} is hard to compute without having some additional auxiliary information beyond what is necessary to compute f.

3. **Classical cryptosystem** (also called a **private key cryptosystem** or a **symmetrical cryptosystem**), we mean a cryptosystem in which, once the enciphering information is known, the deciphering transformation can be implemented in approximately the same order of magnitude of time as the enciphering transformation.

4. A common way to sign a document is with the help of a **hash function**. Roughly speaking, a hash function is an easily computable map $f : x \mapsto h$ from a very long input x to a much shorter output h (for example, from strings of about 10^6 bits to strings of 150 or 200 bits) that has the following property: *it is not computationally feasible to find two different inputs x and x' such that $f(x') = f(x)$.*

5. Deterministic encryption has two disadvantages: (1) if an eavesdropper knows that the plaintext message belongs to a small set (for example, the message is either "yes" or "no"), then she can simply encrypt all possibilities in order to determine which is the supposedly secret message; and (2) it seems to be very difficult to *prove* anything about the security of a system if the encryption is deterministic.

Check your progress 5.2

1. Each user first chooses two extremely large prime numbers p and q (say, of about 100 decimal digits each), and sets $n = pq$. Knowing the factorization of n, it is easy to compute $\varphi(n) = (p - l)(q - 1) = n + 1 - p - q$. Next, the user randomly chooses an integer e between 1 and $\varphi(n)$ which is prime to $\varphi(n)$.

2. A computer program that generates a sequence of digits in a way that no one could duplicate or predict, and which is likely to have all of the statistical properties of a truly random sequence.

Exercise 5.

1. $\binom{m}{2} = m(m-1)/2$ for classical; m for public key; 499500 versus 1000 when $m = 1000$.

2. (a) BH A 2AUCAJEARO; (b) $2047 = 23 \cdot 89$ (see Example) $d_A = 411$; (c) since $\varphi(23)$ and $\varphi(89)$ have small least common multiple 88, any inverse of 179 modulo 88 will work as d_A (e.g., 59).

3. n_A is the product of the Mersenne prirne 8191 and the Fermat prime 65537 - a flamboyantly bad choice; $d_A = 201934721$; "DUMPTHE-STOCK."

4. (a) STOP PAYMENT; (b) (i) 6043; (ii) $n = 113 \cdot 191$.

5. It suffices to prove that $a^{de} \equiv a \mod p$ for any integer a and each prime divisor p of n. This is obvious if $p|a$; otherwise use Fermat's Little Theorem (Proposition 2.1.8).

6. If $m/2 = (p-l)/2 \mod p-1$, then $a^{m/2} \equiv \left(\frac{a}{p}\right)$, which is $+1$ half the time and -1 half the time. In case (ii), use the Chinese Remainder Theorem to show that the probability that an element in $(\mathbf{Z}/n\mathbf{Z})^*$ is a residue modulo p and the probability that it is a residue modulo q are independent of one another, i.e., the situation in case (ii) is like two independent tosses of a coin.

References:

1. Neal Koblit, A course in Number Theory and Cryptography, Springer - Verlag, New York, 2nd edition, 2002.

Suggested Reading:

1. I. Niven and H. S. Zuckermann, An Introduction to Theory of Numbers (Edition 3), Wiley Eastern Ltd, New Delhi 1976

2. D. M. Burton, Elementary Number Theory, Brown Publishers, Iowa, 1989

3. K. Ireland and M. Rosen, A classic Introduction to Modern Number Theory, Springer - Verlag, 1972

4. N. Koblit, Algebraic Aspects of Cryptography, Springer-Verlag, 1998.