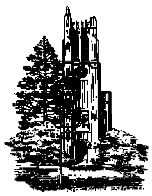
Wednesday June 15, 2016
Lecture 18

Number Theory



Notables

Homework 10 Augmented ; Due Thursday June 16

1. Using the method discussed in class, convert (long hand)
   a. Decimal (that is, base 10) 8888 to binary
   b. Decimal 2555 to base 6
   c. Decimal 8990 to base 19; use symbols 0-9 and A-Z
   You must show ALL your steps to receive full credit.
2. Convert $(260.260)_9$ to base 10.
3. Carry out the following additions in the given base:
   a. $(5665)_7 + (4664)_7 = (\quad)_7$
   b. $(AB56)_{16} + (9868)_{16} = (\quad)_{16}$
4. Page 255, Problem 26
5. Page 256, Problem 30(d), and Problem 48(d)
6. Page 272, Problem 4(f), 21(c), 24(b), and 40(e)
7. Page 284, Problem 6(b)
8. Page 285, Problem 12
9. Page 285, Problem 34

Euclid's Lemma

- Let p be a prime number, and n and m be integers. The following statement are true
  - If $p|n$ and $p|$m then $p|nm$
  - If $p \nmid n$ and $p \nmid m$ then $p \nmid nm$
  - If $p \nmid n$ and $p|nm$ then $p|m$
- Proof is done using Bezout's identity that if x and y are relatively prime, them $rx + sy = 1$ for some integers r and s.

Example

- Find integer x such that:
  - $x \equiv 2 \pmod 3$
  - $x \equiv 3 \pmod 5$
  - $x \equiv 2 \pmod 7$
    - Solution: $x = 23$
    - How?

Chinese Remainder Theorem

Let $m_1, m_2, \ldots, m_n$ be pairwise *relatively prime* positive integers.

The system

$x \equiv a_1 (\bmod m_1)$

$x \equiv a_2 (\bmod m_2)$

$\vdots$

$x \equiv a_n (\bmod m_n)$

has a unique solution modulo $m = m_1 m_2 \ldots m_n$.

That is, there is a *unique* solution x with $0 \le x < m$, and all the other solutions are congruent modulo m to this solution.

How to solve congruence system

Let $m_1, m_2, \ldots, m_n$ be pairwise *relatively prime* positive integers.

The solution x to the system

$x \equiv a_1 (\bmod m_1)$

$x \equiv a_2 (\bmod m_2)$

$\vdots$

$x \equiv a_n (\bmod m_n)$

can be found as follows:

1. Compute $m = m_1 m_2 \cdots m_n$
2. Compute $M_k = \dfrac{m}{m_k} \quad k = 1, 2, \ldots, n$
3. For each M_k find its invers y_k mod m_k, that is, $M_k y_k \equiv 1 \pmod{m_k}$
4. $x \equiv a_1 M_1 y_1 + a_2 M_2 y_2 + a_3 M_3 y_3 + \cdots + a_n M_n y_n \pmod m$

Example

$x \equiv a_1 (\bmod m_1) \quad \Rightarrow \quad x \equiv 2 \pmod 3$
$x \equiv a_2 (\bmod m_2) \quad \Rightarrow \quad x \equiv 3 \pmod 5$
$x \equiv a_3 (\bmod m_3) \quad \Rightarrow \quad x \equiv 2 \pmod 7$

To find x, do the following:

1. Compute $m = m_1 m_2 \cdots m_n \Rightarrow m = 3 \times 5 \times 7 = 105$
2. Compute $M_k = \frac{m}{m_k} \quad k = 1, 2, \ldots, n \Rightarrow M_1 = \frac{105}{3} = 35, M_2 = \frac{105}{5} = 21, M_3 = \frac{105}{7} = 15,$
3. For each M_k find its invers y_k module m_k
   $y_1 = 2$, because $2 \times 35 = 70 \equiv 1 \pmod 3$
   $y_2 = 1$, because $1 \times 21 = 21 \equiv 1 \pmod 5$
   $y_3 = 1$, because $1 \times 15 = 15 \equiv 1 \pmod 7$
4. $x \equiv a_1 M_1 y_1 + a_2 M_2 y_2 + a_3 M_3 y_3 + \cdots + a_n M_n y_n \Rightarrow$
   $x = 2 \times 35 \times 2 + 3 \times 21 \times 1 + 2 \times 15 \times 1 = 233 \equiv 23 \pmod{105}$

Fermat's Little Theorem…

- Theorem: If p is prime and n is an integer not divisible by p, then we have:
  $n^{p-1} \equiv 1 \pmod p$
  Further, when p is prime, for any integer n we have
  $n^p \equiv n \pmod p$
- *Example:*
  - $P = 17, n = 21$
  - $21^{17-1} = 14305686902419853283321 \equiv 1 \pmod{17}$
  - $21^{17} = 300419424950816918947 41 \equiv 4 \pmod{17} = 21 \textbf{ mod } 17$
- Note that the above may not hold when p is not prime
- $7^{4-1} \equiv 3 \ (mod\ 4)$, and $7^4 \equiv 1 \pmod 4$

Proof outline of Fermat's Little Theorem

- p is prime, and n is positive and not divisible by p
- If we take sequence $n, 2n, 3n, \ldots, (p-1)n$ and reduce each one module p, we get a rearrangement of the sequence $1, 2, 3, \ldots, (p-1)$
- Thus
  $n \times 2n \times 3n \times \cdots \times (p-1)n \equiv 1 \times 2 \times 3 \times \cdots \times (p-1) \pmod p$
  $n^{p-1}(p-1)! \equiv (p-1)! \pmod p$
  $n^{p-1} \equiv 1 \pmod p$

Example: Compute 5^{2003} (mod 7)

- Solution 1:
  - 2003 is 11111010011 in binary
  - We compute
    - $5^1 \pmod 7$, which is 5
    - $5^2 \pmod 7$ which is 4
    - $5^{16} \pmod 7$ which is 2
    - $5^{64} \pmod 7$ which is 2
    - $5^{128} \pmod 7$ which is 4
    - $5^{256} \pmod 7$ which is 2
    - $5^{512} \pmod 7$ which is 4
    - $5^{1014} \pmod 7$ which is 2
    - $5 \times 4 \times 2 \times 2 \times 4 \times 2 \times 4 \times 2 \pmod 7$, which is 3

Example: Compute 5^{2003} (mod 7)

- Solution 2: Using Fermat's Little Theorem:
  - By Fermat's Little Theorem we know that:
  - $5^6 \equiv 1 \pmod 7$
  - $5^{1998} = (5^6)^{333} \equiv 1^{333} \equiv 1 \pmod 7$
  - $5^{2003} = 5^5 \times 5^{1998} = 3125 \times 1 \equiv 3 \pmod 7$
  - $5^{2003} \pmod 7 = 3$

Example

- Use Fermat's Little Theorem to compute $5^{2003} \pmod{13}$
- Solution:
  - By Fermat's Little Theorem we know that:
  - $5^{12} \equiv 1 \pmod{13}$
  - $5^{1992} = (5^{12})^{166} \equiv 1^{166} \equiv 1 \pmod{13}$
  - $5^{2003} = 5^{11} \times 5^{1992} = 48828125 \times 1 \equiv 8 \pmod{13}$
  - $5^{2003} \pmod 7 = 8$

Applications of Congruence

- Hashing Functions
- Pseudorandom Numbers
  - Linear congruential method
- Cryptology
  - Caesar cipher
- Check digit(s) in data encoding
  - ISBN
  - Credit Cards

547

Generating Random Numbers

- Applications
  - Networking (Ethernet), Simulations, …
- Issues:
  - What makes a sequence random?
    - There is no "pattern" to it
    - Items are statistically independent
    - The past occurrences will in no way determine the future occurrences. The key is *uncertainty* in the sequence.
      - How can one test?
  - Distribution
    - Uniform, Normal, etc

548

Back to Generating Random Numbers

- Example
  - 14159 26535 89793 23846 26433 83279 50288 41972 ….
  - These are digits in the expansion of π.
- It has long been conjectured that digits in the expansion of π is a very good source of pseudo-random numbers, a conjecture that has still not been proved.
- In 1852 an English mathematician named William Shanks published 527 digits of, and then in 1873 another 180 digits for a total of 707. These numbers were studied statistically, and an interesting excess of the number 7 was observed in the last 180 digits.
- In 1945 von Neumann wanted to study statistical properties of the sequence of digits and used one of the early computers to calculate π to several thousand of places. Fortunately for Shanks his triumph was not spoiled during his lifetime, but his last 180 digits were in error and his last 20 years of effort were wasted. Also there was no "excess of 7s".
- The number has now been calculated to many billions of places

549

Pseudorandom Numbers…

- Linear congruential method
  - Choose: modulus m, multiplier a, increment c, and seed x_0, with $2 \le a < m$, $0 \le c$, $x_0 < m$
  - Generate the sequence $\{x_n\}$ as follows

    $x_{n+1} = (a\, x_n + c) \bmod m$.
- Example: $m = 9$, $a = 7$, $c = 4$, $x_0 = 3$.
  - $x_1 = (7\, x_0 + 4) \bmod 9 = (7*3 + 4) \bmod 9 = 7$
  - $x_2 = (7\, x_1 + 4) \bmod 9 = (7*7 + 4) \bmod 9 = 8$

    ….

  3, 7, 8, 6, 1, 2, 0, 4, 5, 3, 7, 8, 6, 1, 2, 0, 4, 5

550

Pseudorandom Numbers

- In using

  $x_{n+1} = (a\, x_n + c) \bmod m$
  - It is desirable to have a full period, that is, all the numbers 0..m-1 should appear before a repeat.
  - Try:
    - $x_{n+1} = (4\, x_n + 3) \bmod 8$ with seed 2, and you get 2, 3, 7, 7, …. Which is not a full period.
  - When m is a prime number there is a good chance of having a full period.

551

Pseudorandom Numbers…

- Example:

  $x_{n+1} = 7^5\, x_n \bmod (2^{31} - 1)$

  is commonly used. The sequence repeats after

  $2^{31} - 2 = 2{,}147{,}483{,}646$ numbers.

- $2^{31} - 1$ is the largest 32-bit prime.

552

Cryptology

- Caesar's *encryption* process:
  - Represent each letter by an integer from 0 to 25
  - Replace a letter represented by n by the letter represented by, for example, $f(n) = (n + 3) \bmod 26$.
  - Example
    - $M \to 12$, $f(12) = (12+3) \bmod 26 = 15 \to P$
    - "Meet you in the park" is replaced by "Phhw brx lq wkh sdun"
- *Decryption*: To recover the original message, use the inverse function $f^{-1}(n) = (n - 3) \bmod 26$.

553

Cryptology….

- Caesar cipher can be generalized:
  - *Shift cipher*:
    - $f(n) = (n + k) \bmod 26$.
  - *Affine transformation*:
    - $f(n) = (an + b) \bmod 26$, where a and b are integers chosen so that f is a bijection.
    - Example: $f(n) = (7n + 3)$

554

Use of "check" digits in encoding

- Check digits are used for error detection
  - Parity bits used in memories, etc
    - Set the parity bit so that the total number of 1s is even.
    - It detects odd number of errors
  - Last digit in ISBN
    - International Standard Book Number
    - A 10-digit number, $a_1, a_2, \ldots, a_9, a_{10}$,
    - a_{10}, is the check digit, selected such that
      - $a_1+2a_2+3a_3+ \ldots +9a_9+ 10a_{10} \equiv 0 \pmod{11}$
      - The last digit could be 10 and if so, it is displayed as x.
      - It detects single digit errors as well as transposition of two consecutive digits.

555

Example: ISBN

- Consider the following ISBN
  - 0-13-096445-?
  - What should the check digit be?
    - x

556

557

Credit Card Numbers

Card Type	Prefix(es)	Active	Length	Validation	Symbol for coverage chart
American Express	34, 37[1]	Yes	15[2]	Luhn algorithm	AmEx
Bankcard[3]	5610, 560221-560225	No	16	Luhn algorithm	BC
China Union Pay	622 (622126-622925)	Yes	16,17,18,19	*unknown*	CUP
Diners Club Carte Blanche	300-305	Yes	14	Luhn algorithm	DC-CB
Diners Club enRoute	2014, 2149	No	15	no validation	DC-eR
Diners Club International[4]	36	Yes	14	Luhn algorithm	DC-Int
Diners Club US & Canada[5]	55	Yes	16	Luhn algorithm	DC-UC
Discover Card[6]	6011, 65	Yes	16	Luhn algorithm	Disc
JCB	35	Yes	16	Luhn algorithm	JCB
JCB	1800,2131	Yes	15	Luhn algorithm	JCB
Maestro (debit card)	5020,5038,6304,6759	Yes	16,18	Luhn algorithm	Maes
MasterCard	51-55	Yes	16	Luhn algorithm	MC
Solo (debit card)	6334, 6767	Yes	16,18,19	Luhn algorithm	Solo
Switch (debit card)	4903,4905,4911,4936,564182,633110,6333,6759	Yes	16,18,19	Luhn algorithm	Swch
Visa	4[1]	Yes	13,16[7]	Luhn algorithm	Visa
Visa Electron	417500,4917,4913	Yes	16	Luhn algorithm	Visa

558

Luhn's Algorithm

- Consider a 16-digit credit card number, including the check digits, where the numbers are indexed from right to left starting with 0. That is, check digit is in position 0, next digit is in position 1, and so on.
- For each number *k* in an odd position, do the following
  - Multiply *k* by 2, and if the result is ≥ 10, add the digits
  - Add all these numbers to the digits in even positions, and let the sum be *S*
  - The *check digit* = 10 - *s*%10.
  - In other words, the check digit when added to the sum gives a number which is a multiple of 10

559

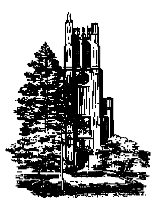
Luhn's Algorithm

P	**15**	14	**13**	12	**11**	10	**9**	8	**7**	6	**5**	4	**3**	2	**1**	0		
N	5	4	9	0	1	2	3	4	5	6	7	8	9	1	2	C		
*2	10		18		2		6		10		14		18		4			
	1	4	9	0	2	2	6	4	1	6	5	8	9	1	4			**62**
																		Total
												C = 10 – (62%10) = 8						

560

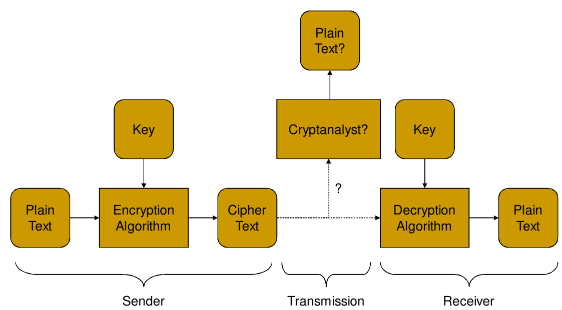
Encryption Potpourri

An application of prime numbers and modular arithmetic



561

Encryption Motivation



562

Symmetric vs Non-Symmetric

- Symmetric
  - Encryption Key = Decryption Key
  - Must Keep Key Secret
  - Good for "Long"" Messages
- Non-Symmetric
  - Encryption Key ≠ Decryption Key
  - Can Publicize One Key (Called "Public Key Cryptography")
  - Good for "Short" Messages

563

RSA (Rivest, Shamir, Adleman)

- Relies on *hardness* of finding prime factorization.
- The Public Key is just a number *k* which is the product of two primes numbers.
- The Private Keys are three other numbers related to the factors of *k*.
- Heavy math is involved here.
  - Chinese Remainder Theorem
  - Fermat's Little Theorem

564

RSA Public key system

- The Public Key is public to everyone!
- Sender encrypts using the Public Key
- Only receiver knows how to decrypt
- Currently, it is practically impossible for the Public to use the Public Key to decrypt

565

RSA

- Key Lengths (Bits)
  - 512 Common
  - 1,024 Recommended For Normal Security
  - 2,048 Recommended For High Security
- Depends on Large Primes
- Was Patented, Expired 2000

566

Public Key Cryptography

- RSA Encryption:
  - Take two large prime numbers p and q. Let $n = p \times q$, and $z = (p-1)(q-1)$.
  - Need an exponent e, which is relatively prime to $z = (p-1)(q-1)$; that is, gcd $(e, z) = 1$.
  - To encrypt an integer message M, we compute $C = M^e$ (**mod** n).
  - So, the Public Key pair are (n, e)
- Example
  - Consider the Public Key pair (n=55 , e=19)
  - To encrypt message $M = 26$, we compute $C = 26^{19}$ (**mod** 55) = 36

567

Public Key Cryptography

- RSA Decryption
  - Need the decryption key d, where $de \equiv 1$ (**mod** z)
  - Need to compute $C^d = (M^e)^d = M^{1+k(p-1)(q-1)}$ (**mod** n)
  - Using Fermat's theorem and the Chinese Remainder Theorem, it can be shown that $C^d = M$ **mod** n.
  - The Private Key pair are ($n = pq$, d)

568

Example

- Consider the Public Key pair (n=55 , e=19)
- To encrypt message $M = 26$, we compute $C = 26^{19}$ **mod** 55 = 36
- The encrypted message 36 is sent.
- To decrypt, use the Private Key (55=11×5 , d = 59)
- Note that $C^{59} = 36^{59}$ **mod** 55 = 26.
- Let $z = (11 - 1)(5 - 1) = 40$. Note gcd (19 , z) = 1
- Also, note that $19 \times 59 \equiv 1$ (**mod** z)

569

Public Key Cryptography, recap

- RSA Encryption:
  - Take two large prime numbers p and q. Let $n = p \times q$, and $z = (p-1)(q-1)$.
  - Need an exponent e, which is relatively prime to $z = (p-1)(q-1)$; that is, gcd $(e, z) = 1$.
  - To encrypt an integer message M, we compute $C = M^e$ **mod** n.
- RSA Decryption
  - Need the decryption key d, where $de \equiv 1$ **mod** z
  - Need to compute C^d **mod** n
  - Using Fermat's theorem and the Chinese Remainder Theorem, it can be shown that $C^d = M$ **mod** n.

570

Example 2

- Consider the Public Key pair ($n = 55$, $e = 19$) and the Private Key pair ($55=11\times5$, $d = 59$)
- To encrypt message $M = 81$, we compute $C = 81^{19}$ **mod** $55 = 36$
- The encrypted message $C = 36$ is sent.
- To decrypt the received message $C = 36$, we compute $C^{59} = 36^{59}$ **mod** $55 = 26$
- But $26 \neq 81$.
- *We did not get the original message back. Why?*
- *Note that* $26 \equiv 81 \pmod{55}$
- Message M should be < n

571

RSA (Special Case)

- RSA Encryption:
  - Need two large prime numbers p and q. Let $n = pq$ and let $z = (p-1)(q-1)$.
  - Need an exponent e, which is relatively prime to $z = (p-1)(q-1)$; that is, gcd $(e, z) = 1$.
  - We will choose $e = 3$. To make sure that gcd $(3, z) = 1$, we will choose p and q such that p mod $3 = 2$ and q mod $3 = 2$.
  - To encrypt an integer message M , we compute $C = M^3$ **mod** n.
  - So, the Public Key pair are (n , 3); and if everyone is using 3, there is no need to announce it!

572

RSA (Special Case) ….

- RSA Decryption
  - Need the decryption key d, where $de \equiv 1$ **mod** z, where $z = (p-1)(q-1)$.
  - In this case $e = 3$, and choosing $d = (2z + 1)/3$ we get an integer and also $de \equiv 1$ **mod** z
  - Need to compute $C^d =$ **mod** n
  - Using Fermat's theorem and the Chinese Remainder Theorem, it can be shown that $C^d = M$ **mod** n.
  - The Private Key is $d = (2(p-1)(q-1) + 1)/3$

573

RAS (special case, recap)

1. Select two different prime numbers p and q such that
   $p \bmod 3 = 2$ and $q \bmod 3 = 2$
2. Compute $d = \dfrac{2(p-1)(q-1)+1}{3}$
3. The Public Key is n, where $n = pq$
4. The Private Keys are p, q, and d
5. To encrypt "number" M compute $C = M^3 \bmod n$
6. To decrypt C, compute $M = C^d \bmod n$

574

RSA, Special Case, Example

- $p = 5$
- $q = 11$
  - Note that 5 **mod** $3 = 2$, and 11 **mod** $3 = 2$
  - This gives $k = 55$, and $d = 27$
- Number to be encrypted $M = 4$
- Encrypted $C = 4^3$ **mod** $55 = 9$
- Decrypted $M = 9^{27}$ **mod** $55 = 4$
  - $4^3 = 64$
  - $9^{27} = 58149737003040059690390169$

575

Sample Public Key

- 6nfX01TUfFaliu1wit5RJ5JQNFBzxWSePsviIml PKReIFSjpktWW6RbGk4pNj+fqh2DOWquaMz dXI27YFVuFJQ==
- This is a number in base 64, using the following symbols
  - 0-25 is 'A'-'Z'
  - 26-51 is 'a'-'z'
  - 52-61 is '0'-'9'
  - 62 is '+'
  - 63 is '/'
  - Pad is '='

576