CS1382 Discrete Computational Structures

Lecture 05: Number Theory and Applications

Spring 2019

Richard Matovu

TEXAS TECH UNIVERSITY

References

The materials of this presentation is mostly from the following:

- Discrete Mathematics and Its Applications (Text book and Slides)

  By Kenneth Rosen, 7th edition

CS1382 Discrete Computational Structures

Integer Representations

Spring 2019

Richard Matovu

TEXAS TECH UNIVERSITY

Representations of Integers

- In the modern world, we use ***decimal,*** or ***base* 10**, *notation* to represent integers.

  For example when we write 965, we mean $9 \cdot 10^2 + 6 \cdot 10^1 + 5 \cdot 10^0$.

- We can represent numbers using any base b, where b is a positive integer greater than 1.

- The bases $b = 2$ (*binary*), $b = 8$ (*octal*) , and $b = 16$ (*hexadecimal*) are important for computing and communications

- The ancient Mayans used base 20 and the ancient Babylonians used base 60.

Base *b* Representations

We can use positive integer b greater than 1 as a base, because of this theorem:

Theorem 1: Let b be a positive integer greater than 1. Then if n is a positive integer, it can be expressed uniquely in the form:

$$n = a_k b^k + a_{k-1} b^{k-1} + \ldots. + a_1 b + a_0$$

where k is a nonnegative integer, $a_0, a_1, \ldots.\ a_k$ are nonnegative integers less than b, and $a_k \neq 0$.

The a_j, $j = 0, \ldots, k$ are called the base-b digits of the representation.

The representation of n given in Theorem 1 is called the ***base b expansion of n*** and is denoted by $(a_k a_{k-1} \ldots. a_1 a_0)_b$.

We usually omit the subscript 10 for base 10 expansions.

Binary Expansions

Most computers represent integers and do arithmetic with binary (base 2) expansions of integers. In these expansions, the only digits used are 0 and 1.

- Examples:
  - What is the decimal expansion of the integer that has $(1\ 0101\ 1111)_2$ as its binary expansion?
  - ***Solution:*** $(1\ 0101\ 1111)_2 = 1\cdot2^8 + 0\cdot2^7 + 1\cdot2^6 + 0\cdot2^5 + 1\cdot2^4 + 1\cdot2^3 + 1\cdot2^2 + 1\cdot2^1 + 1\cdot2^0 = 351.$
  - What is the decimal expansion of the integer that has $(11011)_2$ as its binary expansion?
  - ***Solution:*** $(11011)_2 = 1\cdot2^4 + 1\cdot2^3 + 0\cdot2^2 + 1\cdot2^1 + 1\cdot2^0 = 27.$

Octal Expansions

The octal expansion (base 8) uses the digits { 0,1,2,3,4,5,6,7 }.

Examples:

- What is the decimal expansion of the number with octal expansion $(7016)_8$?
- **Solution**: $7{\cdot}8^3 + 0{\cdot}8^2 + 1{\cdot}8^1 + 6{\cdot}8^0 = 3598$
- What is the decimal expansion of the number with octal expansion $(111)_8$?
- **Solution**: $1{\cdot}8^2 + 1{\cdot}8^1 + 1{\cdot}8^0 = 64 + 8 + 1 = 73$

Hexadecimal Expansions

The hexadecimal expansion needs 16 digits, but our decimal system provides only 10. So letters are used for the additional symbols.

The hexadecimal system uses the digits {0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F}. The letters A through F represent the decimal numbers 10 through 15.

Examples:

- What is the decimal expansion of the number with hexadecimal expansion $(2AE0B)_{16}$?
- **Solution**: $2\cdot16^4 + 10\cdot16^3 + 14\cdot16^2 + 0\cdot16^1 + 11\cdot16^0 = 175627$
- What is the decimal expansion of the number with hexadecimal expansion $(E5)_{16}$?
- **Solution**: $14\cdot16^1 + 5\cdot16^0 = 224 + 5 = 229$

Base Conversion

To construct the base b expansion of an integer n:

- Divide n by b to obtain a quotient and remainder.

  $n = bq_0 + a_0 \quad 0 \leq a_0 \leq b$

- The remainder, a_0 , is the rightmost digit in the base b expansion of n. Next, divide q_0 by b.

  $q_0 = bq_1 + a_1 \quad 0 \leq a_1 \leq b$

- The remainder, a_1, is the second digit from the right in the base b expansion of n.

- Continue by successively dividing the quotients by b, obtaining the additional base b digits as the remainder. The process terminates when the quotient is 0.

continued →

Base Conversion

Example: Find the octal expansion of $(12345)_{10}$

Solution: Successively dividing by 8 gives:

- $12345 = 8 \cdot 1543 + 1$
- $1543 = 8 \cdot 192 + 7$
- $192 = 8 \cdot 24 + 0$
- $24 = 8 \cdot 3 + 0$
- $3 = 8 \cdot 0 + 3$

The remainders are the digits from right to left yielding $(30071)_8$.

Comparison of Hexadecimal, Octal, and Binary Representations

TABLE 1 Hexadecimal, Octal, and Binary Representation of the Integers 0 through 15.

Decimal	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Hexadecimal	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Octal	0	1	2	3	4	5	6	7	10	11	12	13	14	15	16	17
Binary	0	1	10	11	100	101	110	111	1000	1001	1010	1011	1100	1101	1110	1111

Initial 0s are not shown

Each octal digit corresponds to a block of 3 binary digits.

Each hexadecimal digit corresponds to a block of 4 binary digits.

So, conversion between binary, octal, and hexadecimal is easy.

Conversion Between Binary, Octal, and Hexadecimal Expansions

Example: Find the octal and hexadecimal expansions of $(11\ 1110\ 1011\ 1100)_2$.

Solution:

- To convert to octal, we group the digits into blocks of three $(011\ 111\ 010\ 111\ 100)_2$, adding initial 0s as needed. The blocks from left to right correspond to the digits 3,7,2,7, and 4.

  Hence, the solution is $(37274)_8$.

- To convert to hexadecimal, we group the digits into blocks of four $(0011\ 1110\ 1011\ 1100)_2$, adding initial 0s as needed. The blocks from left to right correspond to the digits 3,E,B, and C.

  Hence, the solution is $(3EBC)_{16}$.

Binary Modular Exponentiation

In cryptography, it is important to be able to find b^n **mod** m efficiently, where b, n, and m are large integers.

Use the binary expansion of n, $n = (a_{k-1},\ldots,a_1,a_o)_2$, to compute b^n .

Example:

Compute 3^{11} using this method.

Solution: Note that $11 = (1011)_2$ so that $3^{11} = 3^8\ 3^2\ 3^1$

$$= ((3^2)^2)^2\ 3^2\ 3^1 = (9^2)^2 \cdot 9 \cdot 3 = (81)^2 \cdot 9 \cdot 3 = 6561 \cdot 9 \cdot 3 = 177{,}147.$$

CS1382 Discrete Computational Structures

Primes and Greatest Common Divisors

Spring 2019

Richard Matovu

TEXAS TECH UNIVERSITY

Primes

A positive integer p greater than 1 is called **prime** if the only positive factors of p are 1 and p.

A positive integer that is greater than 1 and is not prime is called **composite**.

Example:

- The integer 7 is prime because its only positive factors are 1 and 7
- But 9 is composite because it is divisible by 3

P = {2,3,5,7,11,15,1719,23,29,31,37,41,43,47,53,59,61,67,71,73,79,83,89, 97}

The Fundamental Theorem of Arithmetic

Theorem:

Every positive integer greater than 1 can be written uniquely as a prime or as the product of two or more primes where the prime factors are written in order of non-decreasing size.

Examples:

- $100 = 2 \cdot 2 \cdot 5 \cdot 5 = 2^2 \cdot 5^2$
- $641 = 641$
- $999 = 3 \cdot 3 \cdot 3 \cdot 37 = 3^3 \cdot 37$
- $1024 = 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 = 2^{10}$

Greatest Common Divisor

Let a and b be integers, not both zero. The largest integer d such that $d \mid a$ and also $d \mid b$ is called the **greatest common divisor of *a* and *b***. The greatest common divisor of a and b is denoted by **gcd(a,b)**.

One can find greatest common divisors of small numbers by inspection.

Example:

- What is the greatest common divisor of 24 and 36?
- **Solution**: gcd(24, 36) = 12
- What is the greatest common divisor of 17 and 22?
- **Solution**: gcd(17,22) = 1

Greatest Common Divisor

The integers a and b are ***relatively prime*** if their greatest common divisor is 1. Example 17 and 22

Definition:

The integers $a_1, a_2, \ldots, a_n$ are ***pairwise relatively prime*** if $\gcd(a_i, a_j) = 1$ whenever $1 \leq i < j \leq n$.

Examples:

- Determine whether the integers 10, 17 and 21 are pairwise relatively prime.
- **Solution**: Because gcd(10,17) = 1, gcd(10,21) = 1, and gcd(17,21) = 1, 10, 17, and 21 are pairwise relatively prime.
- Determine whether the integers 10, 19, and 24 are pairwise relatively prime.
- **Solution**: Because gcd(10,24) = 2, 10, 19, and 24 are not pairwise relatively prime.

Finding the Greatest Common Divisor Using Prime Factorizations

Suppose the prime factorizations of *a* and *b* are: $a = p_1^{a_1} p_2^{a_2} \ldots p_n^{a_n}$, $b = p_1^{b_1} p_2^{b_2} \ldots p_n^{b_n}$,

where each exponent is a nonnegative integer, and where all primes occurring in either prime factorization are included in both. Then:

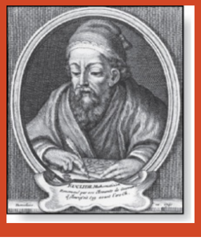
$$\gcd(a,b) = p_1^{\min(a_1,b_1)} p_2^{\min(a_2,b_2)} \ldots p_n^{\min(a_n,b_n)} .$$

Example: $120 = 2^3 \cdot 3 \cdot 5$ $500 = 2^2 \cdot 5^3$

$\gcd(120, 500) = 2^{\min(3,2)} \cdot 3^{\min(1,0)} \cdot 5^{\min(1,3)} = 2^2 \cdot 3^0 \cdot 5^1 = 20$

Finding the gcd of two positive integers using their prime factorizations is **not efficient** because there is no efficient algorithm for finding the prime factorization of a positive integer.

Euclidean Algorithm



Euclid
(325 B.C.E. – 265 B.C.E.)

The Euclidian algorithm is an efficient method for computing the greatest common divisor of two integers. It is based on the idea that gcd(a,b) is equal to gcd(a,c) when $a > b$ and c is the remainder when a is divided by b.

Example: Find gcd(91, 287):

- $287 = 91 \cdot 3 + 14$ — Divide 287 by 91
- $91 = 14 \cdot 6 + 7$ — Divide 91 by 14
- $14 = 7 \cdot 2 + 0$ ← Stopping condition — Divide 14 by 7

gcd(287, 91) = gcd(91, 14) = gcd(14, 7) = 7

continued →

gcds as Linear Combinations

Étienne Bézout
(1730-1783)

Bézout's Theorem:

If a and b are positive integers, then there exist integers s and t such that $\gcd(a,b) = sa + tb$.

Definition:

If a and b are positive integers, then integers s and t such that $\gcd(a,b) = sa + tb$ are called ***Bézout coefficients*** of a and b. The equation $\gcd(a,b) = sa + tb$ is called ***Bézout's identity***.

By Bézout's Theorem, the gcd of integers a and b can be expressed in the form $sa + tb$ where s and t are integers. This is a *linear combination* with integer coefficients of a and b.

- $\gcd(6,14) = (-2)\cdot 6 + 1\cdot 14$

Finding gcds as Linear Combinations

Express gcd(252,198) = 18 as a linear combination of 252 and 198.

Solution: First use the Euclidean algorithm to show gcd(252,198) = 18

i. $252 = 1\cdot 198 + 54$

ii. $198 = 3\cdot 54 + 36$

iii. $54 = 1\cdot 36 + 18$

iv. $36 = 2\cdot 18$

- Now working backwards, from iii and i above
  - $18 = 54 - 1\cdot 36$
  - $36 = 198 - 3\cdot 54$

- Substituting the 2nd equation into the 1st yields:
  - $18 = 54 - 1\cdot(198 - 3\cdot 54) = 4\cdot 54 - 1\cdot 198$
- Substituting $54 = 252 - 1\cdot 198$ (from i)) yields:
  - $18 = 4\cdot(252 - 1\cdot 198) - 1\cdot 198 = 4\cdot 252 - 5\cdot 198$

- This method illustrated above is a two pass method. It first uses the Euclidian algorithm to find the gcd and then works backwards to express the gcd as a linear combination of the original two integers.

- A one pass method, called the ***extended Euclidean algorithm***.

Exercise

Find gcds and find their bézout's identity (express them as linear combinations)

1. gcd(6, 14)
2. gcd(1820, 231)

CS1382 Discrete Computational Structures

Congruences and their Applications

Spring 2019

Richard Matovu

TEXAS TECH UNIVERSITY

Linear Congruences

Definition:

A congruence of the form $a\,x \equiv b\ (\bmod m)$,

where m is a positive integer, a and b are integers, and x is a variable, is called a ***linear congruence***.

The solutions to a linear congruence $ax \equiv b\ (\bmod m)$ are all integers x that satisfy the congruence.

Definition:

An integer $\bar{a}$ such that $\bar{a}\,a \equiv 1 (\bmod m)$ is said to be an ***inverse* of *a* modulo *m***.

Example: 5 is an inverse of 3 modulo 7 since $5 \cdot 3 = 15 \equiv 1 (\bmod 7)$

One method of solving linear congruences makes use of an inverse $\bar{a}$, if it exists. Although we can not divide both sides of the congruence by a, we can multiply by $\bar{a}$ to solve for x.

Inverse of a modulo m

The following theorem guarantees that an inverse of a modulo m exists whenever a and m are relatively prime. Two integers a and b are relatively prime when $\gcd(a,b) = 1$.

Theorem: If a and m are relatively prime integers and $m > 1$, then an inverse of a modulo m exists. Furthermore, this inverse is unique modulo m. (This means that there is a unique positive integer $\bar{a}$ less than m that is an inverse of a modulo m and every other inverse of a modulo m is congruent to $\bar{a}$ modulo m.)

Proof: Since $\gcd(a,m) = 1$, there are integers s and t such that $sa + tm = 1$.

- Hence, $sa + tm \equiv 1 \pmod{m}$.
- Since $tm \equiv 0 \pmod{m}$, it follows that $sa \equiv 1 \pmod{m}$
- Consequently, s is an inverse of a modulo m.

◀

Finding Inverses

The Euclidean algorithm and Bézout coefficients gives us a systematic approaches to finding inverses.

Example: Find an inverse of 3 modulo 7.

Solution: Because $\gcd(3,7) = 1$, by Theorem 1, an inverse of 3 modulo 7 exists.

- Using the Euclidian algorithm: $7 = 2\cdot 3 + 1$.
- From this equation, we get $-2\cdot 3 + 1\cdot 7 = 1$, and see that -2 and 1 are Bézout coefficients of 3 and 7.
- Hence, -2 is an inverse of 3 modulo 7.
- Also every integer congruent to -2 modulo 7 is an inverse of 3 modulo 7, i.e., 5, -9, 12, etc.

Finding Inverses

Example: Find an inverse of 101 modulo 4620.

Solution: First use the Euclidian algorithm to show that gcd(101,4620) = 1.

$4620 = 45\cdot101 + 75$

$101 = 1\cdot75 + 26$

$75 = 2\cdot26 + 23$

$26 = 1\cdot23 + 3$

$23 = 7\cdot3 + 2$

$3 = 1\cdot2 + 1$

$2 = 2\cdot1$

Since the last nonzero remainder is 1, gcd(101,4260) = 1

Working Backwards:

$1 = 3 - 1\cdot2$

$1 = 3 - 1\cdot(23 - 7\cdot3) = -1\cdot23 + 8\cdot3$

$1 = -1\cdot23 + 8\cdot(26 - 1\cdot23) = 8\cdot26 - 9\cdot23$

$1 = 8\cdot26 - 9\cdot(75 - 2\cdot26) = 26\cdot26 - 9\cdot75$

$1 = 26\cdot(101 - 1\cdot75) - 9\cdot75$

$= 26\cdot101 - 35\cdot75$

$1 = 26\cdot101 - 35\cdot(4620 - 45\cdot101)$

$= -35\cdot4620 + 1601\cdot101$

Bézout coefficients : − 35 and 1601

1601 is an inverse of 101 modulo 4620

Using Inverses to Solve Congruences

We can solve the congruence $ax \equiv b \pmod{m}$ by multiplying both sides by $\bar{a}$.

Example: What are the solutions of the congruence $3x \equiv 4 \pmod{7}$.

Solution: We found that -2 is an inverse of 3 modulo 7 (two slides back).

We multiply both sides of the congruence by -2 giving $-2 \cdot 3x \equiv -2 \cdot 4 \pmod{7}$.

Because $-6 \equiv 1 \pmod{7}$ and $-8 \equiv 6 \pmod{7}$, it follows that if x is a solution, then $x \equiv -8 \equiv 6 \pmod{7}$

We need to determine if every x with $x \equiv 6 \pmod{7}$ is a solution. Assume that $x \equiv 6 \pmod{7}$. By Theorem 5 of Section 4.1, it follows that $3x \equiv 3 \cdot 6 = 18 \equiv 4 \pmod{7}$ which shows that all such x satisfy the congruence.

The solutions are the integers x such that $x \equiv 6 \pmod{7}$, namely, 6, 13, 20 … and $-1, -8, -15, \ldots$

Fermat's Little Theorem

Pierre de Fermat
(1601-1665)

Theorem: (*Fermat's Little Theorem*)

If p is prime and a is an integer not divisible by p, then $a^{p-1} \equiv 1 \pmod{p}$

Furthermore, for every integer a we have $a^{p} \equiv a \pmod{p}$

Fermat's little theorem is useful in computing the remainders modulo p of large powers of integers.

Example: Find 7^{222} **mod** 11.

By Fermat's little theorem, we know that $7^{10} \equiv 1 \pmod{11}$, and so $(7^{10})^{k} \equiv 1 \pmod{11}$, for every positive integer k.

Therefore, $7^{222} = 7^{22 \cdot 10 + 2} = (7^{10})^{22} 7^{2} \equiv (1)^{22} \cdot 49 \equiv 5 \pmod{11}$.

Hence, 7^{222} **mod** $11 = 5$.

CS1382 Discrete Computational Structures

Applications of Congruences

Spring 2019

Richard Matovu

TEXAS TECH UNIVERSITY

Hashing Functions

A ***hashing function h*** assigns memory location $h(k)$ to the record that has k as its key.

- A common hashing function is $h(k) = k \textbf{ mod } m$, where m is the number of memory locations.
- Because this hashing function is onto, all memory locations are possible.

Example:

Let $h(k) = k \textbf{ mod } 111$. This hashing function assigns the records of customers with social security numbers as keys to memory locations in the following manner:

$h(064212848) = 064212848 \textbf{ mod } 111 = 14$

$h(037149212) = 037149212 \textbf{ mod } 111 = 65$

$h(107405723) = 107405723 \textbf{ mod } 111 = 14$, but since location 14 is already occupied, the record is assigned to the next available position, which is 15.

Hashing Functions

The hashing function is not one-to-one as there are many more possible keys than memory locations.

When more than one record is assigned to the same location, we say a ***collision*** occurs. Here a collision has been resolved by assigning the record to the first free location.

For collision resolution, we can use a *linear probing function*:

$$h(k,i) = (h(k) + i) \textbf{ mod } m, \text{ where } i \text{ runs from 0 to } m - 1.$$

There are many other methods of handling with collisions. You may cover these in a later CS course.

Pseudorandom Numbers

Randomly chosen numbers are needed for many purposes, including computer simulations.

Pseudorandom numbers are not truly random since they are generated by systematic methods.

The *linear congruential method* is one commonly used procedure for generating pseudorandom numbers.

Four integers are needed: the *modulus* m, the *multiplier* a, the *increment* c, and *seed* x_0, with $2 \leq a < m$, $0 \leq c < m$, $0 \leq x_0 < m$.

We generate a sequence of pseudorandom numbers $\{x_n\}$, with $0 \leq x_n < m$ for all n, by successively using the recursively defined function

$$x_{n+1} = (a\,x_n + c) \textbf{ mod } m.$$

If pseudorandom numbers between 0 and 1 are needed, then the generated numbers are divided by the modulus, x_n / m.

Pseudorandom Numbers

Find the sequence of pseudorandom numbers generated by the linear congruential method with modulus $m = 9$, multiplier $a = 7$, increment $c = 4$, and seed $x_0 = 3$.

Solution: Compute the terms of the sequence by successively using the congruence

$$x_{n+1} = (7x_n + 4) \textbf{ mod } 9, \text{ with } x_0 = 3.$$

$x_1 = 7x_0 + 4 \textbf{ mod } 9 = 7 \cdot 3 + 4 \textbf{ mod } 9 = 25 \textbf{ mod } 9 = 7,$

$x_2 = 7x_1 + 4 \textbf{ mod } 9 = 7 \cdot 7 + 4 \textbf{ mod } 9 = 53 \textbf{ mod } 9 = 8,$

$x_3 = 7x_2 + 4 \textbf{ mod } 9 = 7 \cdot 8 + 4 \textbf{ mod } 9 = 60 \textbf{ mod } 9 = 6,$

$x_4 = 7x_3 + 4 \textbf{ mod } 9 = 7 \cdot 6 + 4 \textbf{ mod } 9 = 46 \textbf{ mod } 9 = 1,$

$x_5 = 7x_4 + 4 \textbf{ mod } 9 = 7 \cdot 1 + 4 \textbf{ mod } 9 = 11 \textbf{ mod } 9 = 2,$

$x_6 = 7x_5 + 4 \textbf{ mod } 9 = 7 \cdot 2 + 4 \textbf{ mod } 9 = 18 \textbf{ mod } 9 = 0,$

Pseudorandom Numbers *(cont…)*

$x_7 = 7x_6 + 4 \textbf{ mod } 9 = 7 \cdot 0 + 4 \textbf{ mod } 9 = 4 \textbf{ mod } 9 = 4,$

$x_8 = 7x_7 + 4 \textbf{ mod } 9 = 7 \cdot 4 + 4 \textbf{ mod } 9 = 32 \textbf{ mod } 9 = 5,$

$x_9 = 7x_8 + 4 \textbf{ mod } 9 = 7 \cdot 5 + 4 \textbf{ mod } 9 = 39 \textbf{ mod } 9 = 3.$

The sequence generated is 3,7,8,6,1,2,0,4,5,3,7,8,6,1,2,0,4,5,3,…

It repeats after generating 9 terms.

Commonly, computers use a linear congruential generator with increment $c = 0$. This is called a ***pure multiplicative generator***. Such a generator with modulus $2^{31} - 1$ and multiplier $7^5 = 16{,}807$ generates $2^{31} - 2$ numbers before repeating.

CS1382 Discrete Computational Structures

Cryptography

Spring 2019

Richard Matovu

Caesar Cipher

- Julius Caesar created secret messages by **shifting** each letter **three letters forward** in the alphabet (sending the last three letters to the first three letters.) For example, the letter B is replaced by E and the letter X is replaced by A.

  This process of making a message secret is an example of **encryption**.

- Here is how the encryption process works:

  - Replace each letter by an integer from Z_{26}, that is an integer from 0 to 25 representing one less than its position in the alphabet.

  - The encryption function is **$f(p) = (p + 3) \bmod 26$**. It replaces each integer p in the set $\{0,1,2,\ldots,25\}$ by $f(p)$ in the set $\{0,1,2,\ldots,25\}$.

  - Replace each integer p by the letter with the position $p + 1$ in the alphabet.

Caesar Cipher - Example

- Encrypt the message “MEET YOU IN THE PARK” using the Caesar cipher.

  - Solution:

    12 4 4 19   24 14 20   8 13   19 7 4   15 0 17 10.

    Now replace each of these numbers p by f(p) = (p + 3) mod 26.

    15 7 7 22   1 17 23   11 16   22 10 7   18 3 20 13.

    Translating the numbers back to letters produces the encrypted message

    “PHHW BRX LQ WKH SDUN.”

Caesar Cipher

- To recover the original message, use $f^{-1}(p) = (p-3)$ **mod** 26.

  So, each letter in the coded message is shifted back three letters in the alphabet, with the first three letters sent to the last three letters.

- This process of recovering the original message from the encrypted message is called **decryption**.

- The Caesar cipher is one of a family of ciphers called **shift ciphers**.

  Letters can be shifted by an integer k, with 3 being just one possibility.

  - The encryption function is $f(p) = (p + k) \bmod 26$
  - The decryption function is $f^{-1}(p) = (p-k) \bmod 26$
  - The integer k is called a **key**.

Shift Cipher - Example

- Encrypt the message "STOP GLOBAL WARMING" using the shift cipher with k = 11.
  - Solution:

    Replace each letter with the corresponding element of Z_{26}.

    18 19 14 15   6 11 14 1 0 11   22 0 17 12 8 13 6.

    Apply the shift $f(p) = (p + 11) \bmod 26$, yielding

    3 4 25 0   17 22 25 12 11 22   7 11 2 23 19 24 17.

    Translating the numbers back to letters produces the ciphertext

    "DEZA RWZMLW HLCXTYR."

Shift Cipher

- Decrypt the message “LEWLYPLUJL PZ H NYLHA ALHJOLY” that was encrypted using the shift cipher with k = 7.

  - Solution:

    Replace each letter with the corresponding element of Z_{26}

    11 4 22 11 24 15 11 20 9 11   15 25   7   13 24 11 7 0   0 11 7 9 14 11 24.

    Shift each of the numbers by $-k = -7$ modulo 26, yielding

    4 23 15 4 17 8 4 13 2 4   8 18   0   6 17 4 0 19   19 4 0 2 7 4 17.

    Translating the numbers back to letters produces the decrypted message

    “EXPERIENCE IS A GREAT TEACHER.”

Block Ciphers

- Ciphers that replace each letter of the alphabet by another letter are called **character** or **mono alphabetic** ciphers. They are vulnerable to cryptanalysis based on letter frequency.

- Block ciphers avoid this problem, by replacing blocks of letters with other blocks of letters.

- A simple type of block cipher is called the **transposition cipher**.
  The key is a permutation σ of the set {1,2,…,m}, where m is an integer, that is a one-to-one function from {1,2,…,m} to itself.

- To encrypt a message, split the letters into blocks of size m, adding additional letters to fill out the final block. We encrypt $p_1,p_2,\ldots,p_m$ as $c_1,c_2,\ldots,c_m = p_{\sigma(1)},p_{\sigma(2)},\ldots,p_{\sigma(m)}$.

- To decrypt the $c_1,c_2,\ldots,c_m$ transpose the letters using the inverse permutation σ^{-1}.

Block Ciphers

Using the transposition cipher based on the permutation σ of the set {1,2,3,4} with $\sigma(1) = 3$, $\sigma(2) = 1$, $\sigma(3) = 4$, $\sigma(4) = 2$,

a. Encrypt the plaintext PIRATE ATTACK

b. Decrypt the ciphertext message SWUE TRAEOEHS, which was encrypted using the same cipher.

Solution:

a. Split into four blocks PIRA TEAT TACK.

Apply the permutation σ giving IAPR ETTA AKTC.

b. σ^{-1} : $\sigma^{-1}(1) = 2$, $\sigma^{-1}(2) = 4$, $\sigma^{-1}(3) = 1$, $\sigma^{-1}(4) = 3$.

Apply the permutation σ^{-1} giving USEW ATER HOSE.

Split into words to obtain USE WATER HOSE.

Cryptosystems

Definition: A *cryptosystem* is a five-tuple (P,C,K,E,D), where

- P is the set of plaintext strings,
- C is the set of ciphertext strings,
- K is the *key space* (set of all possible keys),
- E is the set of encryption functions, and
- D is the set of decryption functions.

- The encryption function in E corresponding to the key k is denoted by E_k and the decryption function in D that decrypts cipher text encrypted using E_k is denoted by D_k.
- Therefore:

$$D_k(E_k(p)) = p, \text{ for all plaintext strings } p.$$

Cryptosystems

Example:

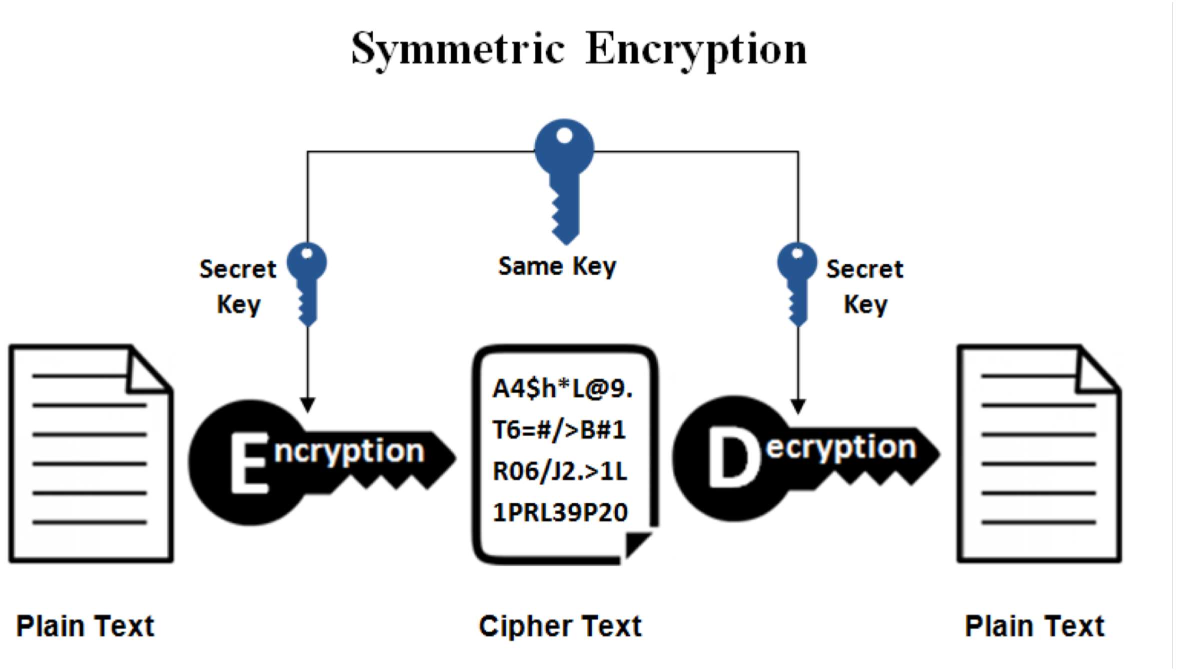
Describe the family of shift ciphers as a cryptosystem.

Solution: Assume the messages are strings consisting of elements in $\mathbf{Z}_{26}$.

- P is the set of strings of elements in $\mathbf{Z}_{26}$,
- C is the set of strings of elements in $\mathbf{Z}_{26}$,
- K = $\mathbf{Z}_{26}$,
- E consists of functions of the form $E_k(p) = (p + k)$ **mod** 26 , and
- D is the same as E where $D_k(p) = (p - k)$ **mod** 26 .

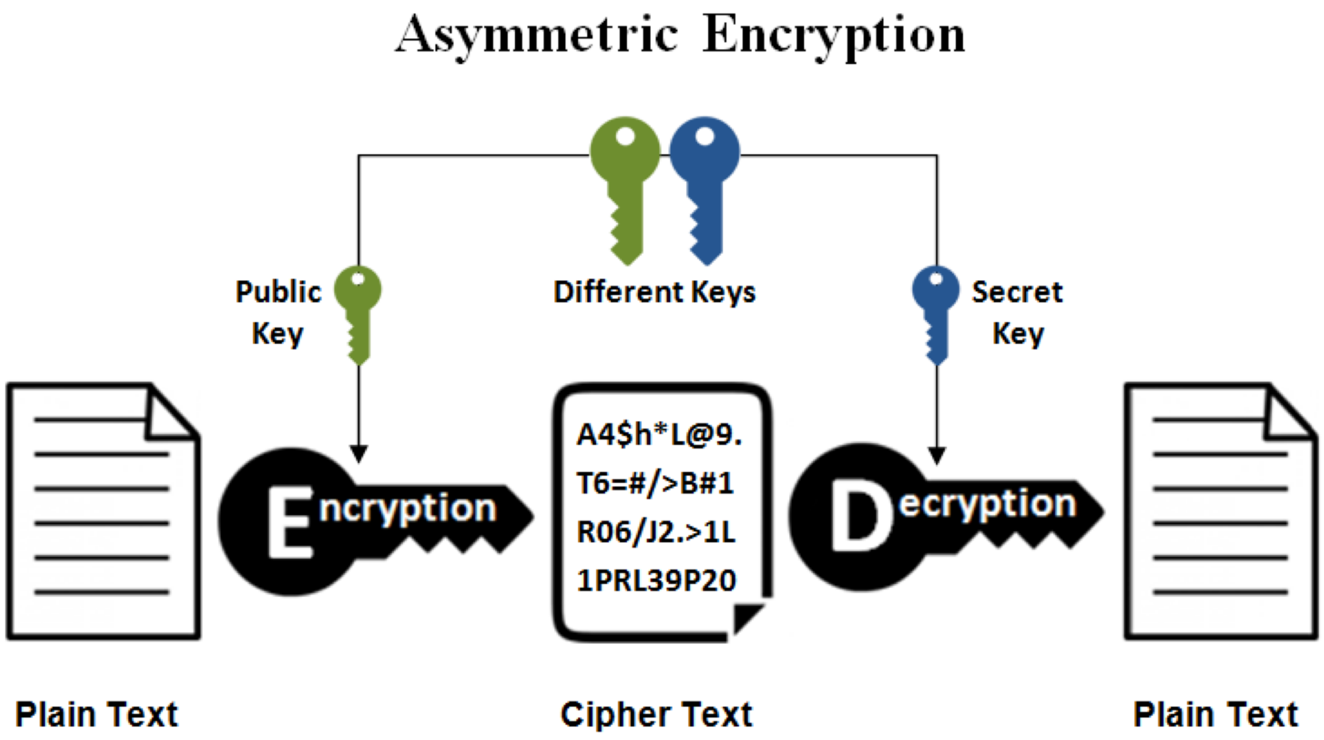
Private Key Cryptography

- All classical ciphers, including shift and affine ciphers, are ***private key cryptosystems***.
- Knowing the encryption key allows one to quickly determine the decryption key
- All parties who wish to communicate using a private key cryptosystem must share the key and keep it a secret.



Public Key Cryptography

- First invented in the 1970s
- Knowing how to encrypt a message does not help one to decrypt the message.
- Everyone can have a publicly known encryption key.
- The only key that needs to be kept secret is the decryption key.



Cryptographic Protocols: Key Exchange

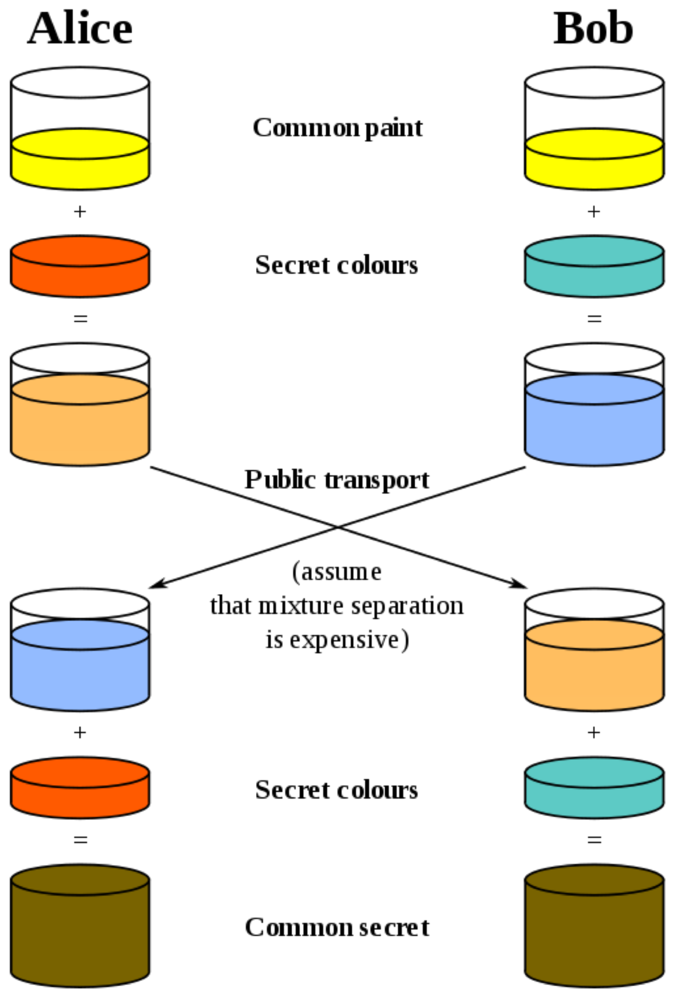
- **Cryptographic Protocols:**

  Exchanges of messages carried out by two or more parties to achieve a particular security goal.

- **Key Exchange:**

  A protocol by which two parties can exchange a secret key over an insecure channel without having any past shared secret information.

Diffie-Hellman Key Exchange Protocol



Here the ***Diffie-Hellman key agreement protocol*** is described by example.

i. Suppose that Alice and Bob want to share a common key.

ii. Alice and Bob agree to use a prime p and a primitive root a of p.

iii. Alice chooses a secret integer k_1 and sends a^{k1} **mod** p to Bob.

iv. Bob chooses a secret integer k_2 and sends a^{k2} **mod** p to Alice.

v. Alice computes $(a^{k2})^{k1}$ **mod** p.

vi. Bob computes $(a^{k1})^{k2}$ **mod** p.

Cryptographic Protocols: Key Exchange

At the end of the protocol, Alice and Bob have their shared key

$$(a^{k2})^{k1} \mathbf{\ mod\ } p = (a^{k1})^{k2} \mathbf{\ mod\ } p.$$

- To find the secret information from the public information would require the adversary to find k_1 and k_2 from a^{k1} **mod** p and a^{k2} **mod** p respectively.
- This is an instance of the discrete logarithm problem, considered to be computationally infeasible when p and a are sufficiently large.

Questions?

Thank You!