Modular arithmetic and cryptography

CSC 1300 – Discrete Structures
Villanova University

Congruence modulo \( m \)

If \( a \) and \( b \) are two integers and \( m \) is a positive integer, then \( a \) is congruent to \( b \) modulo \( m \), denoted \( a \equiv b \pmod{m} \), if

\[
\text{\( a \mod m = b \mod m \),}
\]

i.e. \( a \) and \( b \) have the same remainder when divided by \( m \).

Examples:

\[
\begin{align*}
15 & \equiv 7 \pmod{2} \\
14 & \equiv 2 \pmod{12} \\
15 & \equiv 95 \pmod{10} \\
-6 & \equiv 24 \pmod{2} \\
8 & \equiv -4 \pmod{12}
\end{align*}
\]

Equivalent formulations of congruence modulo \( m \)

\( a \) is congruent to \( b \) modulo \( m \)

- \( a \equiv b \pmod{m} \)
- \( a \mod m = b \mod m \)
- \( a, b \) have the same remainder when divided by \( m \)
- \( a - b \) is divisible by \( m \)
- \( m \) divides \( a - b \) (denoted \( m \mid (a - b) \))
- \( a - b = km \) for some \( k \in \mathbb{Z} \)

Equivalence Relations

\( a \equiv b \pmod{m} \) is an equivalence relation.

A relation on a set \( A \) is called an equivalence relation if it is

(i) reflexive,
(ii) symmetric,
(iii) transitive.

Formalizes the notion of “equivalence” or “sameness”.

Examples of equivalence relations:

- “to be equal” (for numbers or sets)
- “to have the same number of elements” (for sets)
- “to have the same age”
- “to have the same remainder after division by 2” (i.e., parity)
- “to have the same first 3 bits” (for bit strings)
- “to have the same truth table” (for propositions)
Equivalence classes

Let $R$ be an equivalence relation on a set $X$, and let $a$ be an element of $X$. The set of all elements of $X$ that are related to $a$ by $R$ is called the equivalence class for $a$ and is denoted by $[a]$. Any element $b \in [a]$, $b$ is called a representative of $[a]$.

Example: Consider the relation “... is congruent to ... modulo 2” on the set $Z$. What is $[4]$?

Example: Let $X$ be the set of all bit strings of length at least 3, and $R$ be the relation “agree in the first three bits”. Find

- $[001]$
- $[1101101]$
- $[0011]$

Partitions

Theorem. Let $R$ be an equivalence relation on a set $X$. Then the equivalence classes of $R$ form a partition of $X$.

A partition of a set $X$ is a collection of disjoint nonempty subsets of $X$ that have $X$ as their union.

The collection $X_1, X_2, X_3, X_4$ is a partition of $X$.

Useful facts about congruences

The following hold for $n \geq 1$:

- $a \equiv a \pmod{n}$
- $a \equiv b \pmod{n} \Rightarrow b \equiv a \pmod{n}$
- $(a \equiv b \pmod{n} \land b \equiv c \pmod{n}) \Rightarrow a \equiv c \pmod{n}$
- $a \equiv b \pmod{n} \Rightarrow a + c \equiv b + c \pmod{n}$
- $a \equiv b \pmod{n} \Rightarrow ac \equiv bc \pmod{n}$
- $a \equiv b \pmod{n} \Rightarrow a^k \equiv b^k \pmod{n}$
- $(a \equiv b \pmod{n} \land c \equiv d \pmod{n}) \Rightarrow a + c \equiv b + d \pmod{n}$
- $(a \equiv b \pmod{n} \land c \equiv d \pmod{n}) \Rightarrow ac \equiv bd \pmod{n}$
Application: Cryptography

- Encoding – convert letters to numeric codes
- Encryption: Apply numeric transformations that obscure the text.
- Use modular arithmetic to:
  - "contain" numeric codes within specific bounds (e.g., 0-25)
  - exploit properties of number theory to create transformations that are difficult to reverse.
- Private key cryptography
- Public key cryptography

Private Key Cryptography – Shift ciphers

- Caesar cipher: Shift each letter by 3
- Example: Encoding
  - Plaintext: boyfriend
  - Convert: 01 14 24 05 17 08 04 13 03 25
  - Add 3: 04 17 27 08 20 11 07 16 06 28
  - ...mod 26: 04 17 01 08 20 11 07 16 06 02
  - Ciphertext: erbiulhqgc

- Decoding...

Private Key Cryptography – One-time pads

- Secret “one-time-pad” of randomly generated keys
- Example: Encoding
  - Plaintext: boyfriend
  - Convert: 01 14 24 05 17 08 04 13 03 25
  - pad: 23 02 00 19 20 13 07 15 06 18
  - add key: 24 16 24 24 37 21 11 28 09 43
  - ...mod 26: 24 16 24 24 11 21 11 02 09 17
  - Ciphertext: yqyyylvcljr

- Decoding...

Public Key Cryptography

- Eliminates the need to deliver a private key
- Two keys: one for encoding, one for decoding
- Known algorithm
  - security based on security of the decoding key – note, no key delivery problem
- Essential element:
  - knowing the encoding key will not reveal the decoding key
Effective Public Key Encryption

- Encoding method E and decoding method D are inverse functions on message M:
  \[ D(E(M)) = M \]
- Computational cost of E, D reasonable
- D cannot be determined from E, the algorithm, or any amount of plaintext attack with any computationally feasible technique
- E cannot be broken without D (only D will accomplish the decoding)
- Any method that meets these criteria is a valid Public Key Encryption technique

It all comes down to this:

- key used for decoding is dependent upon the key used for encoding, but the relationship cannot be determined in any feasible computation or observation of transmitted data

RSA Algorithm (Rivest, Shamir, Adelman)

- Choose 2 large prime numbers, \( p \) and \( q \), each more than 100 digits
- Compute \( n = p \times q \) and \( z = (p-1)(q-1) \)
- Choose \( d \), relatively prime to \( z \)
- Find \( e \), such that \( e \times d \equiv 1 \pmod{z} \)
  - i.e., \( e \times d \mod z = 1 \)
  - This produces \( e \) and \( d \), the two keys that define the E and D methods.

Public Key encoding

- Convert \( M \) into a bit string
- Break the bit string into blocks, \( P \), of size \( k \)
  - \( k \) is the largest integer such that \( 2^k < n \)
  - \( P \) corresponds to a binary value: \( 0 < P < n \)
- Encoding method
  - \( E = \text{Compute } C = P^e \mod n \)
- Decoding method
  - \( D = \text{Compute } P = C^d \mod n \)
- \( e \) and \( n \) are published (public key)
- \( d \) is closely guarded and never needs to be disclosed
An example:

- Choose 2 large prime numbers, p and q, each more than 100 digits (Well, not so big for our example)
  - p=7; q=11;
- Compute n = p*q and z = (p-1)*(q-1)
  - n = 77; z = 60
- Choose d, relatively prime to z (60, in this case)
  - Could we choose 9? 8? (Why not?)
  - Choose d = 13 (List some other possible choices)
- Find e, such that e*d \equiv 1 \pmod z
  - What multiple of d, when divided by z, gives a remainder of 1?
  - e=37; We have our keys: e, d, and n are required. No amount of trying will find d if given e and n.

Using the keys

- We need to choose a size (k) for the blocks of code that will be sent together.
  - k is the largest integer such that 2^k < n
  - Our n is 77, what is k?
    - 2^3 is 8; 2^4 is 16; 2^5 is 32
    - k is 5
- How will we encode our characters? ASCII? Other?
  - Let’s keep it simple. A = 1, B = 10, C = 11, etc.
    - How many bits to we need to encode all 26 letters?
    - 26 < 32. 32 is 2^5. We need 5 bits to encode all 26 letters
    - So, A = 00001, B=00010, C=00011, D=00100,
    - What is S?

Encode a message

- Let’s encode “CAT”
- First, convert the letters to binary
  - C = 00011   A = 00001   T = 10100
  - Our message, in binary, is
    - 000110000110100
- Group the bits in blocks of 6 (because k is 6)
  - 000000 110000 110100
  - Note extra 0s needed to fill the block. Add extra 0s at the left so that the numeric value does not change

Encoding, continued

- We are sending this: 000000 110000 110100
  - Each 6-bit block is a chunk of plaintext, P
  - Each block of code must be encrypted.
    - Our e is 37, n is 77
      - P^{37} \pmod 77 is the encrypted version of our plaintext block
      - 0^{37} is 0;
      - 110000 is 48. 48^{37} \pmod 77 is 27
      - 110100 is 52. 52^{37} \pmod 77 is 24
  - We transmit the binary version of 0 27 24 (000000 011011 011000)
Your turn

- You receive this: 000000011011011000
  - There will not be any convenient spacing between the blocks in the transmitted message. That is not necessary.
- It has been encoded with your public key. You have the private key, d = 13 n = 77
- Show the decoding process

Another exercise

- Transmit and send a three letter message. Use the same e, d, n that we have developed here.
- Receive a message from someone else and decode it.

A practical note

- There is a lot more to security than encryption.
- Encryption coding is done by a few experts
- Understanding how the common encryption algorithms work is useful in choosing the right approach for your situation.
- Our interest here is in providing assurance that access to protected resources will be limited to those with legitimate rights.

Issues

- Intruder vulnerability
  - If an intruder intercepts a request from A for B’s public key, the intruder can masquerade as B and receive messages from A intended for B. The intruder can send those same or different messages to B, pretending to be A.
  - Prevention requires authentication of the public key to be used.
- Computational expense
Digital Signatures

• Some messages do not need to be encrypted, but they do need to be authenticated: reliably associated with the real sender
  — Protect an individual against unauthorized access to resources or misrepresentation of the individual’s intentions
  — Protect the receiver against repudiation of a commitment by the originator

Digital Signature basic technique

Sender

Intention to send

E(Random Number)

where E is A’s public key

Message and

D(E(Random Number))

Receiver

Public key encryption with implied signature

• Add the requirement that E(D(M)) = M
• Sender A has encoding key E_A, decoding key D_A
• Intended receiver has encoding (public) key E_B.
• A produces E_B(D_A(M))
• Receiver calculates E_A(D_B(E_B(D_A(M))))
  — Result is M, but also establishes that only A could have encoded M

Digital Signature Standard (DSS)

• Verifies that the message came from the specified source and also that the message has not been modified
• More complexity than simple encoding of a random number, but less than encrypting the entire message
• Message is not encoded. An authentication code is appended to it.
Digital Signature – SHA (Secure Hash Algorithm)

Encryption summary

- **Problems**
  - Intruders can obtain sensitive information
  - Intruder can interfere with correct information exchange

- **Solution**
  - Disguise messages so an intruder will not be able to obtain the contents or replace legitimate messages with others

Important methods

- **Private Key Encryption**
  - Fast, reasonably good encryption
  - Key distribution problem
  - Can be made secure at considerable cost through use of one-time pad

- **Public Key Encryption**
  - More secure
    - Based on the difficulty of factoring very large numbers
  - No key distribution problem
  - Computationally intense

Legal and ethical issues

- People who work in these fields face problems with allowable exports, and are not always allowed to talk about their work.
- Is it desirable to have government able to crack all codes?
- What is the tradeoff between privacy of law abiding citizens vs. the ability of terrorists and drug traffickers to communicate in secret?