Week 3: Mathematics for Cryptography II

Dr. Qublai K. Ali Mirza

University of Gloucestershire

qalimirza@glos.ac.uk

Overview

UNIVERSITY OF GLOUCESTERSHIRE

Congruence

- Given two integers a and b, we say that a and b are *congruent* with respect to m iff:
  - they both result in the same remainder (aka. `modulo`) r when divided by m
  - their difference $(a - b)$ is *divisible* by m
- To put this into a mathematical equation, it is written as:

$$a \equiv b \ (mod \ m)$$

UNIVERSITY OF GLOUCESTERSHIRE

Examples of congruence

- 2 and 12 are congruent with respect to 10 since

$$2 \equiv \ 12 \ (mod \ 10)$$

- Similarly 3 and 6 are congruent with respect to 3 since

$$3 \equiv \ 6 \ (mod \ 3)$$

Properties

- For any two integers a and b, the following properties hold true [1]:
  - $a \equiv a \pmod{m}$ for all values of a
  - If $a \equiv b \pmod{m}$, then $b \equiv a \pmod{m}$
  - If $a \equiv b \pmod{m}$ and If $b \equiv c \pmod{m}$, then $a \equiv c \pmod{m}$

Greatest Common Divisor

- If we have two *non-zero* integers a and b, the greatest common divisor ($gcd\ (a,\ b)$) is the largest number that can divide both of them.
- For example, if $a = 50$ and $b = 10$ then $\gcd(50,\ 10)$ then is 5 since
  - $50 = 5 \times 10 = 5 \times 5 \times 2$
  - $10 = 5 \times 2$

Obtaining the GCD

- To calculate the *gcd* of two integers *a* and *b*, there are two different techniques:
  - *Prime factorisation*
    - Breaking down each number into its multipliers and taking the common multipliers
    - Easy to understand, difficult to implement
  - *Euclidean algorithm*
    - Based on *modulo arithmetic*
    - Can be implemented through programming

Euclidean algorithm

- Based on the observation that for any two nonnegative integers a and b,

$$gcd(a, b) = gcd(b, a \ mod \ b)$$

- Used in the factorisation of large-scale numbers
- Can be implemented through programming

UNIVERSITY OF GLOUCESTERSHIRE

How it works

$$64 = 1 \times 52 + 12$$

How it works

$$64 = 1 \times 52 + 12$$

$$52 = 4 \times 12 + 4$$

UNIVERSITY OF GLOUCESTERSHIRE

How it works

$$64 = 1 \times 52 + 12$$

$$52 = 4 \times 12 + 4$$

$$12 = 3 \times 4 + 0$$

Algorithm

INPUT: Integers a > b >=0

OUTPUT: gcd(a, b)

1. if b = 0 then return (a);

2. return (gcd(b, a mod b))

Figure: Euclidean algorithm, from [2]

Overview

- If we have a set of congruences such that:

$$\begin{aligned} x &\equiv a_1 \ (mod \ m_1) \\ x &\equiv a_2 \ (mod \ m_2) \\ &.. \\ x &\equiv a_k \ (mod \ m_k) \end{aligned}$$

- There is exactly one solution $x \in Z_m$ satisfying all of them.
- Z_m is a residue set consisting of all possible moduli with respect to a *modm* operation.

Conditions

1. $gcd(m_i, m_j) = 1$
2. $m = m_i \times m_2 \times m_3 .., \times m_k$
3. $a_1, a_2, ..., a_k$ and $m_1, m_2, ..., m_k$ are integers

UNIVERSITY OF GLOUCESTERSHIRE

How it works

- Step 1: For each z_i in $z_1, z_2, ..., z_k$, calculate $z_i = m/m_i$
- Step 2: For each y_i in $z_1, z_2, ..., z_k$, calculate $y_i = z_i^{-1}(mod\ m_i)$
- Step 3: The value of x then becomes $x = a_1 y_1 z_1 + .. + a_k y_k z_k$

Example

Imagine that we want to find the value of x in Z_{60} such that

$$\begin{aligned} x &\equiv 3 \ (mod\ 4) \\ x &\equiv 2 \ (mod\ 3) \\ x &\equiv 4 \ (mod\ 5) \end{aligned}$$

Step 0 - Initialise: $a_1 = 3,\ a_2 = 2,\ a_3 = 4,\ m_1 = 4,\ m_2 = 3,$
$m_3 = 5,\ m = 4 \times 3 \times 5 = 60$

Example (cont.)

Step 1: Calculate $z_1 = m/m_1 = 60/4 = 15$, $z_2 = 20$, and $z_3 = 12$
Step 2: Calculate the module inverse $z_i y_i \equiv 1 \ mod \ m_i$. In other words, we want to solve the following:

$$
\begin{aligned}
15y_1 &\equiv 1 \ mod \ 4 \\
20y_2 &\equiv 1 \ mod \ 3 \\
12y_3 &\equiv 1 \ mod \ 5
\end{aligned}
$$

Based on calculations, $y_1 = 3, y_2 = 2$, and $y_3 = 3$

Example (cont.)

Step 3: Finally calculate the value x by:

$$x \equiv a_1 y_1 z_1 + \ a_2 y_2 z_2 \ + \ a_3 y_3 z_3 \ (\ mod \ 60)$$
$$x \equiv 3 \times 3 \times 15 + 2 \times 2 \times 20 \ + \ 4 \times 3 \times 12 \ (\ mod \ 60)$$
$$x \equiv 59 \ (mod \ 60)$$

UNIVERSITY OF GLOUCESTERSHIRE

Overview

- Measurement of information *uncertainty*
- Developed by Claude Shannon in 1946
- The higher the entropy value is, the greater is the *unpredictability* of the data
- It is calculated using:

$$H(X) = -\sum_{i=1}^{n} P(x_i) \log_b P(x_i)$$

UNIVERSITY OF GLOUCESTERSHIRE

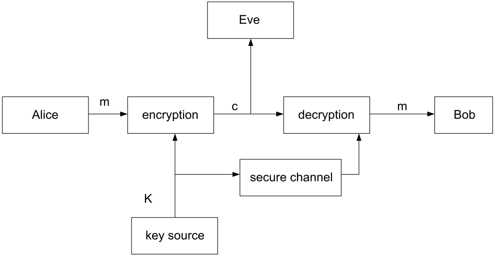

Figure: Cryptography overview, adapted from [3]

Passive attacks [5]

- *Ciphertext only*
  - Analysing the ciphertext produced
- Known plaintext
  - Uses *both* plaintext and ciphertext
- Chosen plaintext
  - Uses plaintext on a given cyphersystem to analyse ciphertext obtained
- Chosen ciphertext
  - Uses ciphertext on a given cyphersystem to analyse plaintext obtained

- According to Shannon, a cryptosystem has *perfect secrecy* iff

$$H(M) \leq H(K)$$

- For this to happen, the size (or more specifically, the length) of K needs to be at least *as large as* the size of the plaintext.

Bringing it together

- Today we looked at congruency and greatest common divisor
- We also looked at Euclidean algorithm and the Chinese remainder theorem
- We discussed entropy and its role within cryptography
- Next week: *Symmetric encryption*

Post-sessional work

- Using [4] as a starting point, discuss what symmetric encryption is and its different variants
- Upload your completed work to *Moodle* before next *Monday*

UNIVERSITY OF GLOUCESTERSHIRE

References

L Lindahl. "Lectures on Number Theory". In: *Sweden: Uppsala University Retrieved http://www2. math. uu. se/~ astrombe/talteori2016/lindahl2002. pdf* (2002).

Wenbo Mao. *Modern cryptography: theory and practice*. Prentice Hall Professional Technical Reference, 2003.

Shannon's Theory of Secrecy. http://www.eit.lth.se/fileadmin/eit/courses/edi051/lecture_notes/LN3.pdf. Accessed: 2018-01-17.

Gustavus J Simmons. "Symmetric and asymmetric encryption". In: *ACM Computing Surveys (CSUR)* 11.4 (1979), pp. 305–330.

Alexander Stanoyevitch. *Introduction to Cryptography with mathematical foundations and computer implementations*.

Q & A