CS 6530 Applied Cryptography

July-Nov 2025

Introduction to Cryptography and Data Security

31st July 2025

Dr. Manikantan Srinivasan

(Material covered is based on Chapter 1 of

Understanding Cryptography – Second Edition

Courtesy: Slides by Authors - Christof Paar and Jan Pelzl)

Contents of Chapter 1

- **Overview on the field of cryptology**
- Basics of symmetric cryptography
- Cryptanalysis
- Substitution Cipher
- Modular arithmetic
- Shift (or Caesar) Cipher and Affine Cipher

Modular Arithmetic

Short Introduction to Modular Arithmetic

Why do we need to study modular arithmetic?

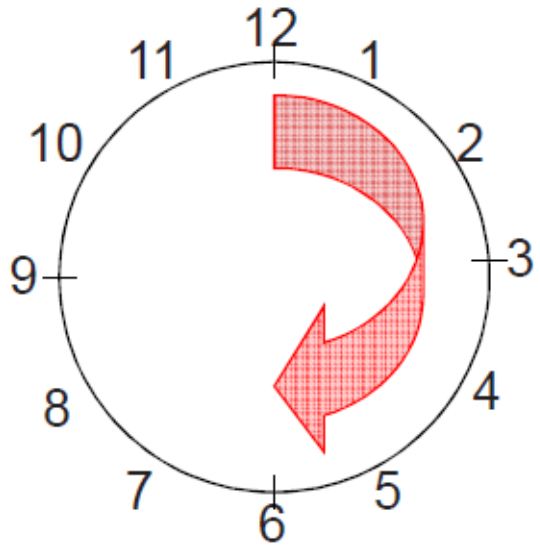
- Extremely important for asymmetric cryptography (RSA, elliptic curves etc.)
- Some historical ciphers can be elegantly described with modular arithmetic (cf. Caesar and affine cipher later on).

Short Introduction to Modular Arithmetic

Generally speaking, most cryptosytems are based on **sets of numbers** that are

1. **discrete** (sets with integers are particularly useful)
2. **finite** (i.e., if we only compute with a finiely many numbers)

Seems too abstract? --- Let's look at a finite set with discrete numbers we are quite familiar with: a clock.



Interestingly, even though the numbers are incremented every hour we never leave the set of integers:

1, 2, 3, … 11, 12, 1, 2, 3, … 11, 12, 1, 2, 3, …:

Short Introduction to Modular Arithmetic

- We develop now an arithmetic system which allows us to **compute** in finite sets of integers like the 12 integers we find on a clock (1,2,3, … ,12).
- It is crucial to have an operation which „keeps the numbers within limits“, i.e., after addition and multiplication they should never leave the set (i.e., never larger than 12)

Definition: Modulus Operation

Let *a, r, m* be integers and $m > 0$. We write

$$a \equiv r \bmod m$$

if (a-r) *is divisable by m.*

- "*m*" is called the **modulus**
- "*r*" is called the **remainder**

Examples for modular reduction.

- Let a= 12 and m= 9 : $12 \equiv 3 \bmod 9$
- Let a= 37 and m= 9: $34 \equiv 7 \bmod 9$
- Let a= -7 and m= 9: $-7 \equiv 2 \bmod 9$

(you should check whether the condition „*m* divides (*r-a*)"holds in each of the 3 cases)

Properties of Modular Arithmetic (1)

- **The remainder is not unique**

It is somewhat surprising that for every given modulus *m* and number *a*, there are (infinitely) many valid remainders.

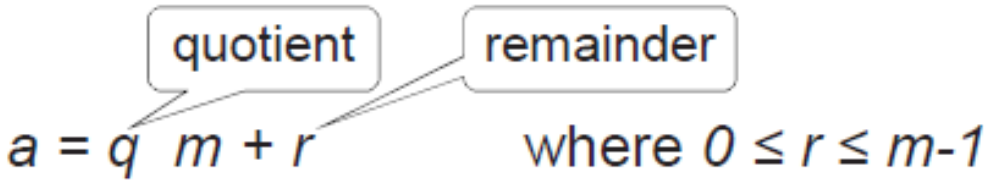
Example:

- $12 \equiv 3 \bmod 9$ → 3 is a valid remainder since 9 divides (12-3)
- $12 \equiv 21 \bmod 9$ → 21 is a valid remainder since 9 divides (12-21)
- $12 \equiv -6 \bmod 9$ → -6 is a valid remainder since 9 divides (12-(-6))

Properties of Modular Arithmetic (2)

- **Which remainder do we choose?**

By convention, we usually agree on the **smallest positive integer *r*** as remainder. This integer can be computed as



- Example: $a=12$ and $m=9$

$$12 = 1 \times 9 + 3 \quad \rightarrow \quad r = 3$$

Remark: This is just a convention. Algorithmically we are free to choose any other valid remainder to compute our crypto functions.

Properties of Modular Arithmetic (3)

- **How do we perform modular division?**

First, note that rather than performing a division, we prefer to multiply by the inverse. Ex:

$$b / a \equiv b \times a^{-1} \bmod m$$

The inverse a^{-1} of a number a is defined such that:

$$a\, a^{-1} \equiv 1 \bmod m$$

Ex: What is $5 / 7 \bmod 9$?

The inverse of $7 \bmod 9$ is 4 since $7 \times 4 \equiv 28 \equiv 1 \bmod 9$, hence:

$$5 / 7 \equiv 5 \times 4 = 20 \equiv 2 \bmod 9$$

- **How is the inverse compute?**

The inverse of a number $a \bmod m$ only exists if and only if:

$$\gcd(a, m) = 1$$

(note that in the example above gcd(5, 9) = 1, so that the inverse of 5 exists modulo 9)

For now, the best way of computing the inverse is to use exhaustive search. In Chapter 6 of *Understanding Cryptography* we will learn the powerful Euclidean Algorithm which actually computes an inverse for a given number and modulus.

Properties of Modular Arithmetic (4)

- **Modular reduction can be performed at any point during a calculation**

Let's look first at an example. We want to compute $3^8 \bmod 7$ (note that exponentiation is extremely important in public-key cryptography).

1. Approach: Exponentiation followed by modular reduction

$$3^8 = 6561 \equiv \mathbf{2} \bmod 7$$

Note that we have the intermediate result 6561 even though we know that the final result can't be larger than 6.

2. Approach: Exponentiation with intermediate modular reduction

$$3^8 = 3^4 3^4 = 81 \times 81$$

At this point we reduce the intermediate results 81 modulo 7:

$$3^8 = 81 \times 81 \equiv 4 \times 4 \bmod 7$$

$$4 \times 4 = 16 \equiv \mathbf{2} \bmod 7$$

Note that we can perform all these multiplications without pocket calculator, whereas mentally computing $3^8 = 6561$ is a bit challenging for most of us.

General rule: For most algorithms it is advantageous to reduce intermediate results as soon as possible.

An Algebraic View on Modulo Arithmetic: The Ring Z_m (1)

We can view modular arithmetic in terms of sets and operations in the set. By doing arithmetic modulo *m* we obtain **the integer ring Z_m** .with the following properties:

- **Closure**: We can add and multiply any two numbers and the result is always in the ring.
- Addition and multiplication are **associative**, i.e., for all $a,b,c \varepsilon Z_m$

  $a + (b + c) = (a + b) + c$

  $a \times (b \times c) = (a \times b) \times c$

  and addition is **commutative**: $a + b = b + a$
- The **distributive law** holds: $a\times(b+c) = (a\times b)+(a\times c)$ for all $a,b,c \varepsilon Z_m$
- There is the **neutral element *0* with respect to addition**, i.e., for all $a \varepsilon Z_m$

  $a + 0 \equiv a \bmod m$
- For all $a \varepsilon Z_m$, there is always an **additive inverse element *–a*** such that

  $a + (-a) \equiv 0 \bmod m$
- There is the **neutral element 1 with respect to multiplication**, i.e., for all $a \varepsilon Z_m$

  $a \times 1 \equiv a \bmod m$
- The **multiplicative inverse a^{-1}**

  $a \times a^{-1} \equiv 1 \bmod m$

  exists only for some, but not for all, elements in Z_m.

An Algebraic View on Modulo Arithmetic: The Ring Z_m (2)

Roughly speaking, a ring is a structure in which we can always add, subtract and multiply, but we can only divide by certain elements (namely by those for which a multiplicative inverse exists).

- We recall from above that an element $a \, \varepsilon \, Z_m$ has a multiplicative inverse only if:

  gcd $(a, m) = 1$

  We say that a is **coprime** or **relatively prime** to m.

- Ex: We consider the ring $Z_9 = \{0,1,2,3,4,5,6,7,8\}$

  The elements 0, 3, and 6 do not have inverses since they are not coprime to 9.

  The inverses of the other elements 1, 2, 4, 5, 7, and 8 are:

  $1^{-1} \equiv 1 \bmod 9$ $\quad$ $2^{-1} \equiv 5 \bmod 9$ $\quad$ $4^{-1} \equiv 7 \bmod 9$

  $5^{-1} \equiv 2 \bmod 9$ $\quad$ $7^{-1} \equiv 4 \bmod 9$ $\quad$ $8^{-1} \equiv 8 \bmod 9$

Shift (or Caesar) Cipher and Affine Cipher

Shift (or Caesar) Cipher (1)

- Ancient cipher, allegedly used by Julius Caesar
- Replaces each plaintext letter by another one.
- Replacement rule is very simple: Take letter that follows after *k* positions in the alphabet

Needs mapping from letters → numbers:

A	B	C	D	E	F	G	H	I	J	K	L	M
0	1	2	3	4	5	6	7	8	9	10	11	12
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25

- Example for *k* = *7*

Plaintext = `ATTACK` = 0, 19, 19, 0, 2, 10

Ciphertext = `haahr` = 7, 0, 0, 7, 9, 17

Note that the letters ”wrap around” at the end of the alphabet, which can be mathematically be expressed as reduction modulo 26, e.g., *19 + 7 = 26 ≡ 0 mod 26*

Shift (or Caesar) Cipher (2)

- Elegant mathematical description of the cipher.

Let k, x, y ε {0,1, …, 25}

- Encryption: $y = e_k(x) \equiv x + k \bmod 26$
- Decryption: $x = d_k(x) \equiv y - k \bmod 26$

- Q; Is the shift cipher secure?
- A: No! several attacks are possible, including:
  - Exhaustive key search (key space is only 26!)
  - Letter frequency analysis, similar to attack against substitution cipher

Affine Cipher (1)

- Extension of the shift cipher: rather than just adding the key to the plaintext, we also multiply by the key
- We use for this a key consisting of two parts: k = (a, b)

Let k, x, y ε {0,1, …, 25}

- Encryption: $y = e_k(x) \equiv a\,x + b \bmod 26$
- Decryption: $x = d_k(x) \equiv a^{-1}(y - b) \bmod 26$

- Since the inverse of *a* is needed for inversion, we can only use values for *a* for which:

$$\gcd(a, 26) = 1$$

There are 12 values for *a* that fulfill this condition.

- From this follows that the key space is only 12 x 26 = 312 (cf. Sec 1.4 in *Understanding Cryptography*)
- Again, several attacks are possible, including:
  - Exhaustive key search and letter frequency analysis, similar to the attack against the substitution cipher

Summary

What we learnt in chapter 1

- Never ever develop your own crypto algorithm unless you have a team of experienced cryptanalysts checking your design.
- Do not use unproven crypto algorithms or unproven protocols.
- Attackers always look for the weakest point of a cryptosystem. For instance, a large key space by itself is no guarantee for a cipher being secure; the cipher might still be vulnerable against analytical attacks.
- Key lengths for symmetric algorithms in order to thwart exhaustive key-search attacks:
  - 64 bit: insecure except for data with extremely short-term value
  - 128 bit: long-term security of several decades, unless quantum computers become available (quantum computers do not exist and perhaps never will)
  - 256 bit: as above, but probably secure against attacks by quantum computers.
- Modular arithmetic is a tool for expressing historical encryption schemes, such as the affine cipher, in a mathematically elegant way.

Thank you – Q&A