Affine Ciphers

CMPUT 396

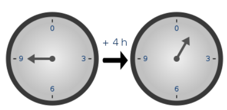
Overview

- The multiplicative cipher is like Caesar but uses multiplication instead of addition.
- The affine cipher combines the multiplicative cipher and the Caesar cipher.
- To understand how it works, we need to review modular arithmetic and factoring.

Modular arithmetic

- Modular arithmetic is a system of arithmetic for integers, where numbers "wrap around" upon reaching a certain value—the modulus.
- E.g. 17 mod 12 = 5
- Python uses '%'

```
x = 17 % 12
```



Factors

- A *factor* (divisor) of an integer n, is an integer m that multiplied by some integer produces n.

  n = m * k
- In this case, n is a *multiple* of m.
- n is *divisible* by m if m is a factor (divisor) of n; that is, dividing n by m leaves no *remainder*.
- E.g., 7 is a factor of 35 because 7 * 5 = 35 and 35 is divisible by (or, *is a multiple of*) 7
- The positive factors of 35 are: 1, 5, 7, 35.

Greatest common divisor

- The greatest common divisor (GCD) of two integers is their largest positive integer that divides each of the integers.
- For example, the GCD of 24 and 30 is 6 because their common factors are: 1, 2, 3, 6.
- Prime numbers have only two factors: 1 and n
- Two numbers are called relatively prime (or coprime) if their GCD equals 1.

Euclid's algorithm for GCD

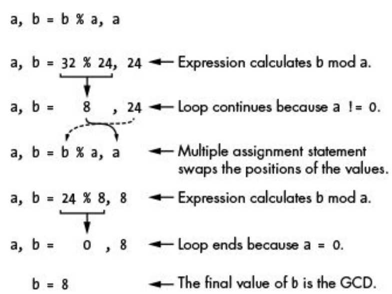
- Note that the GCD of a and b also divides a − b
- Formally:

  GCD(a,b) = GCD(b, a mod b)

  GCD(a,0) = a
- In Python:

```
def gcd(a, b):
  while a != 0
    a, b = b % a, a
  return b
```

How the gcd() function works



The multiplicative cipher

- In the Caesar cipher, you add the key (shift):

  $C_i = (M_i + K) \bmod 26$
- In the multiplicative cipher, you *multiply* the index by the key:

  $C_i = (M_i * K) \bmod 26$
- E.g. if the key is 11, then ‘F’ encrypts as ‘C’

(index(‘F’) * key) mod 26 = (5 * 11) mod 26 = 3

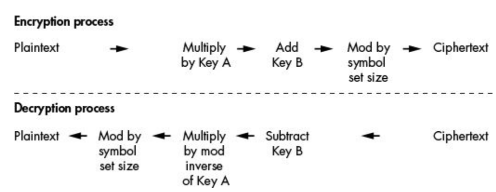
Choosing valid keys

- Not all numbers will work as a key.

  $C_i = (M_i * K) \bmod 26$
- E.g. 5 * 6 mod 26 = 4 = 18 * 6 mod 26 = 4, so both ‘F’ and ‘S’ would encrypt as ‘E’
- The key and the alphabet size must be coprime
- You can use the gcd() function to check this
- Note: ‘A’ encrypts as ‘A’ for any key value

Affine cipher

- The affine cipher has two keys: A and B
- $C_i = ((M_i * A) + B) \bmod 26$
- $M_i = ((C_i - B) * \mathit{modInv}(A)) \bmod 26$



Encrypting with the Affine Cipher

In this example, A = 5 and B = 8

Plaintext	a	f	f	i	n	e
x	0	5	5	8	13	4
5x+8	8	33	33	48	73	28
(5x+8) mod 26	8	7	7	22	21	2
Ciphertext	I	H	H	W	V	C

Modular inverse

- A *modular inverse* of A modulo N is X such that:

  $(X * A) \bmod N = 1$
- E.g. modular inverse of 15 mod 26 is 7
- To fine a modular inverse, use *Euclid’s extended algorithm*
- Note: because A and N cannot be co-prime, the number of different keys is less than N

Cracking the affine cipher (1)

- Suppose we guess that the message starts with "DEAR…" and the first two ciphertext letters are RA
- Replace the letters with their indices to get:

  (3 * A + B) mod 26 = 17

  (4 * A + B) mod 26 = 0
- Subtract one equation from the other:

  ((4 - 3)* A + (B − B)) mod 26 = 0 - 17
- which simplifies to: A mod 26 = -17, so A = 9
- This implies (4 * 9 + B) mod 26 = 0 , so B = 16
- The key is found to be (A = 9, B = 16)

Cracking the affine cipher (2)

- Suppose we guess that S'K enciphers I'M
- Replace the letters with their indices to get:

  (8 * A + B) mod 26 = 18

  (12 * A + B) mod 26 = 10
- Subtraction produces: (4 * A) mod 26 = -8 = 18
- In this case, there are two solutions because 4 and 26 are not co-prime: (A = 11, B = 8) and (A = 24, B = 8)
- We can find the key by guessing another cipher letter.
- e.g. if L enciphers F, then (5 * 11 + 8) mod 26 = 11, but (5 * 24 + 8) mod 26 ≠ 11, so the key is (A = 11, B = 8)