Math for Cryptography I

They Have Played Us For Absolute Fools

@nebu

Outline

- Meta
- Motivation
- Basics
- More Interesting Structures
- Big Theorem 1: Chinese Remainder Theorem
- Big Theorem 2: Fermat's Little Theorem
- RSA

What this Presentation Is

- An introduction to the mathematics that's useful for cryptography.
- A way to put Husnain's excellent presentation on factoring in context.
- Hopefully, give you a better idea of why things like this happen:

What this Presentation Isn't

- Challenge heavy
- Interesting for math people. The later ones in this series probably will be.

By the End

- Not be scared of all the math, and actually *enjoy* crypto.
- Gain a deeper understanding of why mucking around with numbers creates elegant systems that can hide information reliably.

It all ties together at the end, I promise, even though we explore seemingly unrelated ideas.

Outline

Note on Notation

- $\mathbb{N}$: Natural numbers: $\{1, 2, 3, ...\}$
- $\mathbb{Z}$: Integers: $\{..., -3, -2, -1, 0, 1, 2, 3, ...\}$

A Problem

Take a number $n \in \mathbb{N}$. Add its digits to get another number n'. Add the digits of n'. Keep doing this till you get an n that is one digit long and return that n.

- 65536
- 25
- 7

Solution

$$d = \begin{cases} 9, & n \equiv 0 \mod 9 \\ n \mod 9, & \text{otherwise} \end{cases}$$

```
def sumdigit(n: int) -> int:
    if n % 9 == 0:
        return 9
    return n % 9
```

Why Does This Work

Number theory --- stay tuned.

Outline

Divides

- $5 \mid 25$
- $6 \mid 54$
- $7 \nmid 15$

For $a, b \in \mathbb{Z}$ say $a \mid b$ if there exists a $c \in \mathbb{Z}$ such that $b = ac$.

a is called a divisor or factor of b.

Quiz!

True or false:

- $25 \mid 125$
- $45 \mid 900$
- $3 \mid 4$
- $37 \mid 111$
- $7 \mid 1001$

Greatest Common Divisor

For two integers, the greatest common divisor is the largest integer that divides both of them.

For instance, $\gcd(54, 36) = 18$.

How To Compute a GCD

Naively, write out the list of all divisors of both integers, and find the maximum common element in these two arrays.

54: [1, 54, 2, 27, 3, 18, 6, 9]

36: [1, 36, 2, 18, 3, 12, 4, 9, 6]

Technically I've only listed positive divisors.

Programming Problem

How do you:

1. Break a number into its divisors.
2. Find the common maximum of two unsorted arrays.

Solution

1. Refer to the factoring meeting. Spoiler: it's neither fast or easy.
2. Put the first array in a hash table, then go through the second array keeping track of the largest value that's in the table.

Euclidean Algorithm

Brilliant insight based on the idea that $\gcd(a, b) = \gcd(b, r)$ where r is the remainder you get when you divide a by b.

The algorithm is to keep repeating the step $\gcd(a, b)$ with $\gcd(b, r)$ until you get an r that is 0. Once that happens, the previous r is the GCD.

Euclidean Algorithm

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

$$36 = 18 \cdot 1 + 0$$

Euclidean Algorithm

$$54 = 36 \cdot 1 + \boxed{18}$$

$$36 = 18 \cdot 1 + 0$$

18 is our GCD.

Euclidean Algorithm

Don't worry too much about *why* this works --- it is enough to recognize that it does and to understand that computing the GCD is computationally efficient[1] since it's mostly floor division and getting remainders in a loop.

We'll come back to this later.

[1] The exact upper bound is $O(\log b)$.

Outline

What Are Integers Made Of?

The idea of divisibility suggests that integers are 'made of', well, products of other integers.

$$288 = 4 \cdot 72$$

Which are in turn made of other integers.

$$288 = (2 \cdot 2) \cdot (18 \cdot 4)$$

And so on…

What Are Integers Made Of?

At some point you'll reach some numbers you can't split up anymore.

$$288 = 2^5 \cdot 3^2$$

These are the primes, and are the LEGO® blocks of all the integers.

Fundamental Theorem of Arithmetic

States that all integers can be uniquely represented as a product of prime powers. That is, for all $n \in \mathbb{Z}$, n has a unique representation of the form:

$$n = \prod_{i=1}^{k} p_i^{\alpha_i}$$

where all p_i are prime and all $\alpha_i \in \mathbb{N}$.

This is just fancy math talk for the idea that all integers are formed from prime number LEGO® blocks multiplied together.

By The Way…

That $\prod$ thing looks scary, but it's just multiplication in a for loop.

$$\prod_{i=1}^{4} i^2$$

is the same thing as

```
prod = 0
for i in range(1,5):
    prod *= i**2
```

Coprimality

Two numbers are coprime if their greatest common divisor is 1.

Alternately, you can view it as the fact the numbers don't have any LEGO® prime pieces in common.

5 and 8 are coprime, 4 and 16 aren't.

Modularity

When we divide things, we get expressions like

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

This can be written as:

$$54 \equiv 18 \mod 36$$

That is, 54 leaves a remainder of 18 when it's divided by 36.

Modularity

In general, note that

$$a \equiv b \mod m \iff m \mid a - b$$

since $m \mid a - b$

$\iff km = a - b \iff km + b = a \iff a \equiv b \mod m$

for some $k \in \mathbb{Z}$.

Okay, More Quiz!

- $16 \mod 4$
- $77 \mod 10$
- $29 \mod 2$
- $29 \mod 3$

Congruences Are Pretty Powerful

Remember the sum of digits thing? It's based off the following idea. A k digit integer n in base 10 is essentially:

$$n = 10^{k-1} n_{k-1} + 10^{k-2} n_{k-2} + \cdots + 10^1 n_1 + 10^0 n_0$$

Properties of Congruences

Taking this $\mod 9$,

$$n \equiv 10^{k-1} n_{k-1} + 10^{k-2} n_{k-2} + \cdots + 10^1 n_1 + 10^0 n_0 \pmod 9$$

We can take each term $\mod 9$ separately.
It's a bit hard to think about what terms like $10^{k-1} n_{k-1} \mod 9$ are, though. We can do:

$$\underbrace{10 \cdot 10 \cdots 10}_{k-1 \text{ times}} \cdot n_{k-1} \pmod 9$$

We can also take each 10 in this expression $\mod 9$ separately. Then, the term is:

$$\underbrace{1 \cdot 1 \cdots 1}_{k-1 \text{ times}} \cdot n_{k-1} \pmod 9$$

Properties of Congruences

The whole expression reduces to

$$n \equiv n_{k-1} + n_{k-2} + \cdots + n_1 + n_0 \pmod 9$$

This is just the sum of digits of n: which is what we wanted!

It's pretty useful that we can manipulate sums and products that way in congruences.

Linear Congruences

This is where it gets fun. Since we have $\equiv$, which behaves like a weird $=$, we can write a sort of linear equation:

$$16x \equiv 8 \pmod{24}$$

Or the general form,

$$ax = b \pmod{m}$$

Linear Congruences

$$ax = b \pmod{m}$$

We won't go into how to solve these, but we note that for $d = \gcd(a, m)$,

1. If $d \nmid b$, there are no solutions.
2. If $d \mid b$, there are exactly d solutions.[2]

[2] This is left loosely defined in this presentation.

A Special Congruence

Consider the congruence:

$$ax = 1 \mod m$$

By our rules about $\gcd(a, m)$, for a solution to exist, $d \mid 1$.

What divides 1? Just 1, so d needs to be 1 for this equation to be solvable. Since d is 1, there is exactly one solution.

Multiplicative Modular Inverse

$$ax \equiv 1 \pmod{m}$$

For the equation to be solvable, $\gcd(a, m) = 1$, that is, a and m must be coprime.

This solution is unique and is called the multiplicative modular inverse of $a \pmod{m}$, also written

$$x \equiv a^{-1} \pmod{m}$$

MODINV Examples

Consider:

$$5^{-1} \pmod 7$$

```
print(pow(5, -1, 7))
```

```
3
```

This is true, since $5 \cdot 3 \equiv 15 \equiv 1 \pmod 7$.

```
print(pow(4, -1, 8))
```

```
ValueError: base is not invertible for the
given modulus
```

Be careful with your Python superpowers!
Make sure a and m are coprime.

Outline

Systems of Congruences

There are certain things whose number is unknown. If we count them by threes, we have two left over; by fives, we have three left over; and by sevens, two are left over. How many things are there?

This forms the system:

$$\begin{aligned} x &\equiv 2 \pmod 3 \\ x &\equiv 3 \pmod 5 \\ x &\equiv 2 \pmod 7 \end{aligned}$$

CRT gives the solution $x = 23$ or the general solution $x = 105k + 23$, $k \in \mathbb{Z}$. Note that $105 = 3 \cdot 5 \cdot 7$.

Weaker Form of the Chinese Remainder Theorem

We consider a special case of the Chinese Remainder Theorem: a system of two equations

$$x \equiv a_1 \pmod{p}$$

$$x \equiv a_2 \pmod{q}$$

The Chinese Remainder Theorem states that $x_1 \equiv x_2 \pmod{pq}$ for any two solutions x_1 and x_2. The CRT also requires that p and q be coprime.

What We Need to Know

All we care about is the fact that if we have some congruence $\mod pq$ where p and q are coprime, we can separately solve the congruences $\mod p$ and $\mod q$.

Outline

The Fun Begins

This is the heart of the talk, but it took a while to get here…

Fermat's Little Theorem states that for prime p and a coprime to p,

$$a^{p-1} \equiv 1 \pmod{p}$$

We prove this using the method of necklaces (not to be confused with bracelets).

Definitions

- For our purposes, an alphabet is a set of symbols. An alphabet has a length, a, that is the number of symbols it has.
- A string is a group of symbols. Using an alphabet of at least one symbol, we can build infinitely many strings.
- A necklace is a string whose last symbol and first symbol are next to each other: it's a circular linked list.
  - Necklaces can be represented by more than one string. Consider the necklace 'SIGPWNY' is represented by all the strings 'SIGPWNY', 'YSIGPWN', 'NYSIGPW', and so on.

Making an Alphabet and Some Strings

Let's use the alphabet with symbols AB, $a = 2$, $p = 5$. Let the length of the strings we make using this alphabet be a prime p. Then, we can make a^p total strings. Written out, our strings are:

AAAAA, AAAAB, AAABA, AAABB, AABAA, AABAB, AABBA, AABBB, ABAAA, ABAAB, ABABA, ABABB, ABBAA, ABBAB, ABBBA, ABBBB, BAAAA, BAAAB, BAABA, BAABB, BABAA, BABAB, BABBA, BABBB, BBAAA, BBAAB, BBABA, BBABB, BBBAA, BBBAB, BBBBA, BBBBB.

The Question

The question is, for such a set of strings defined by a and p, how many necklaces that use more than one symbol are in the set?

Given that p is prime and $\gcd(a, p) = 1$.

Strings to Necklaces

We need to figure out some sort of intuitive relation between strings and necklaces.

A good place to start is how *many* ways a necklace can be represented as a string. Consider the necklace 'ABBABBABBABB'. The strings we can do are:

- 'ABBABBABBABB': naturally.
- 'BABBABBABBAB': ROT1
- 'BBABBABBABBA': ROT2
- 'ABBABBABBABB': back to square one!

Clearly, there are only three representations of the necklace as a string.

A Useful Observation

The reason that there were only three strings to represent 'ABBABBABBABB' is because 3 divides its length, 12. There is a substring of length 3 in 'ABBABBABBABB', which wraps and therefore covers all possible necklaces.

In general, if S is the length of the necklace, and if $T \mid S$, there will be T possible strings that can represent the necklace.

Another way of looking at this is that the period of the string is T.

Really think about this and ask questions. This is a key part in the proof.

Our Case

In our case, we know S is a prime p. The only T that divides p are 1 and p itself.

That is, there are only two cases: for a prime length, some necklaces are represented uniquely by just one string, and others are represented by p strings.

Case $T = 1$

This is quite obvious, if the period is 1, then this is a one-symbol string like `'AAAAA'` or `'BBBBB'`.

There are exactly a such cases (one case for each symbol).

Case $T = p$

These are the remaining $a^p - a$ cases. In all these cases, each string represents exactly one necklace and each necklace is represented by exactly p strings (since $T = p$).

Answer

So, the answer to the question ``how many necklaces of at least two symbols'' is

$$\frac{a^p - a}{p}$$

Done!

Since there must be an integral number of necklaces, $p \mid a^p - a$.

By the definition of congruence, $p \mid a^p - a \implies a^p \equiv a \pmod{p}$.

Since a is coprime to p, we can write this as $a^{p-1} \equiv 1 \pmod{p}$ and we are done!

A Manual Verification

Remember our alphabet? Here it is split into the sets $T = p = 5$ and $T = 1$:

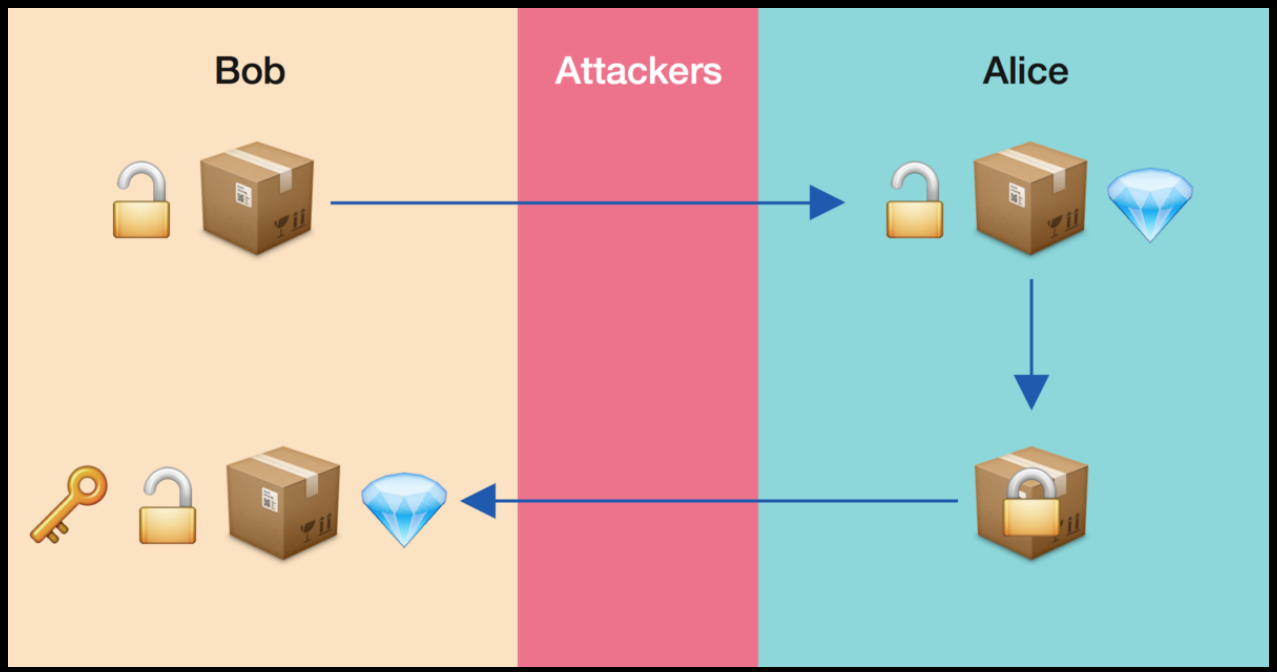
AAAAB	AAABA	AABAA	ABAAA	BAAAA
AAABB	AABBA	ABBAA	BBAAA	BAAAB
AABAB	ABABA	BABAA	ABAAB	BAABA
AABBB	ABBBA	BBBAA	BBAAB	BAABB
ABABB	BABBA	ABBAB	BBABA	BABAB
ABBBB	BBBBA	BBBAB	BBABB	BABBB

AAAAA
BBBBB

Outline

Public Key Cryptography? Help!

Pretty clear:



Mathematical Description: Encrpytion and Decryption

- Alice gets Bob's 'public key', the values (n, e). She also has the message she wants to send Bob, m. She computes

$$c = m^e \pmod{n}$$

  and sends it to Bob.

- Bob has his private key, d. He computes

$$m = c^d \pmod{n}$$

  to recover the message.

It's that simple!

Mathematical Description: Key Generation

Okay, but where do n, e, d come from? Let's take a look.

- Get two primes p and q.
  - Multiply them to get $n = pq$.
  - Compute $\phi(n) = (p-1)(q-1)$.
- Choose any e coprime to $\phi(n)$.
  - Compute $d \equiv e^{-1} \pmod{\phi(n)}$

Public key is (n, e). Private key is d.

Proof

We want to show that $m \equiv c^d \equiv (m^e)^d \pmod{pq}$.

By the way we calculated d, $ed \equiv 1 \pmod{\phi(pq)}$.

Or, $ed - 1 = a(p-1) = b(q-1)$ for some non-negative a and b, since $\phi(pq) = (p-1)(q-1)$.

CRT Part

Using our takeaway from the CRT, we know that to prove $m \equiv m^{ed} \pmod{pq}$ it is sufficient to show $m \equiv m^{ed} \pmod{p}$ and $m \equiv m^{ed} \pmod{p}$ separately. Since the method is identical, we do only p here.

1. If $m \equiv 0 \pmod{p}$, $m^{ed} \equiv 0 \equiv m \pmod{p}$.
2. Otherwise,

$$m^{ed} \equiv m^{ed-1} \cdot m \equiv m^{a(p-1)} \cdot m \pmod{p}$$

Can you figure out where to go from here?

FLT Part

$$\equiv 1^a m \equiv m \pmod{p}$$

by Fermat's Little Theorem. Do the same thing for q using $b(q-1)$, and we have that

$$m \equiv c^d \equiv (m^e)^d \pmod{pq}$$