MATHEMATICAL CIPHERS

Delving into the Cryptographic Complexities of Number Theory and Abstract Algebra

We saw in the prev meet that...

Perfect secrecy

Encryption scheme (Gen, Enc, Dec) with message space M and ciphertext space C is perfectly secret

IF

for every distribution over M, every $m \in M$, and every $c \in C$ with $Pr[C=c] > 0$, it holds that

$$Pr[M = m \mid C = c] = Pr[M = m]$$

But in real world, this is a very stringent condition.

So then what?

Lets say a hacker can test 2^80 keys in an year (a typical desktop can do 2^60 while 2^80 for supercomputers)

And how many years since the big bang? **13.8 billion years** (ie ≈ 2^34)

So a supercomputer since bigbang could have searched for ≈ 2^114 keys

So if I use somewhere near 2^128 keys, then **is it safe enough?**
(Modern key space is 2^128 keys or more)

Computational Secrecy

It means that while an attacker may gain some information about the plaintext from the ciphertext, breaking the encryption scheme is infeasible with current computational resources.

- **Security is based on the computational difficulty of certain problems**
- **It's always possible, though highly unlikely!**

I’ve Got Nothing to Hide

'I've Got Nothing to Hide' and Other Misunderstandings of Privacy

In this short essay, written for a symposium in the San Diego Law Review, Professor Daniel Solove examines the nothing to hide argument. When asked about gover

ssrn.com

[Link to the paper]

"I've Got Nothing to Hide" and Other Misunderstandings of Privacy, written by Daniel J. Solove, challenges common misconceptions about privacy. He argues that privacy is not solely about hiding wrongdoing but encompasses broader values such as autonomy, trust, and individuality. The paper refutes arguments that dismiss privacy concerns, emphasizing its importance for fostering creativity, free expression, and democratic values. Solove advocates for a nuanced understanding of privacy in society, calling for legal and social frameworks that prioritize its protection while balancing competing interests.

!! Do give it a read !!

Some key advances related to this...

1) Digital Personal Data Protection Act (DPDPA), 2023

- The Act grants data principals (individuals) rights such as access, correction, deletion, and the right to opt-out of data processing.
- BUT...The DPDPA has provisions that grant the Indian government certain exemptions, which has raised concerns about potential overreach and the balance between privacy and state surveillance.

2) Information Technology Act, 2000 (IT Act) and Amendments:

- Enhanced Penalties: Recent amendments through the Jan Vishwas Act, 2023, have increased penalties for breaches of data protection rules under the IT Act.

Let’s begin!

Number Theory

Why its needed?

- **RSA cryptosystem –** relies on the difficulty of factoring large composite numbers into their prime factors.
- **Diffie-Hellman Key Exchange**: Uses modular exponentiation.
- **Digital Signature Algorithm (DSA) –** depends on the difficulty of solving the discrete logarithm problem.
- **Error detection** – Many cryptographic hash functions and error detection codes use number theoretic properties to ensure data integrity.

What all do we need to see?

- Efficient algorithms for various computations
- Asymptotic computation time depends on how long the input is rather than its actual value. It means that, input length, lets say $||a|| = O(\log a)$ will decide the computation time rather than the actual value of a.

Modular Arithmetics

- [a mod N] means the remainder is ‘a’ when a number is divided by N.
- Examples. $9 \equiv 4 \bmod 5$, $100 \equiv 1 \bmod 3$ etc.
- What about computation time?

It can be assumed that modular computations take the same time as integer computations, ie both are efficient computations.

- Modular addition / subtraction / multiplication / reduction

Modular Exponentiation

a^b mod N

- Computation time of a^b = ||a^b|| = O(b * ||a||) = O(b * log a)
- How to compute a^b mod N?

```
exp(a, b, N) {
    // assume b ≥ 0
    ans = 1;
    for (i=1, i ≤ b; i++)
        ans = [ans * a mod N];
    return ans;
  }
```

- **Running time? O(b) ?**

Modular Exponentiation

Does a better algorithm exist?

Modular Exponentiation

Does a better algorithm exist?

```
exp(a, b, N) {
    // assume b ≥ 0
    x=a, t=1;
    while (b>0) {
        if (b odd)
            t = [t * x mod N], b = b-1;
        x = [x^2 mod N],   b = b/2; }
    return t; }
```

Running time ? O(log b) ? ie O(|| b||) ?

Modular Inverse

The modular inverse of an integer a modulo n is another integer b such that

$$\mathbf{ab \equiv 1 \bmod n}$$

A modular inverse of a modulo n exists if and only if gcd(a,n) = 1.

- Modular inverses can be found using Extended Euclidean Algorithm.
- Notice, if p is a prime, then 1, 2, 3, 4, …., p-1 are all invertible modulo p.

Modular Inverse

Let $\Phi(N)$: the number of invertible elements modulo N.
$= \{ a \in \{1,2,3...N-1\} : gcd(a,N) = 1\}$

- if N is prime then $\Phi(N) = ?$
- if N=pq, such that p and q are distinct primes, then $\Phi(pq) = ?$

Modular Inverse

Let $\Phi(N)$: the number of invertible elements modulo N.

$= \{ a \in \{1,2,3...N-1\} : \gcd(a,N) = 1\}$

- if N is prime then $\Phi(N) = ?$
- if N=pq, such that p and q are distinct primes, then

$\Phi(pq) = N - 1 - (q - 1) - (p - 1) = N - q - p + 1 = (p-1)(q-1)$

What is $\Phi(91)$?

Fermat's Little Theorem

- If p is a prime number and a is an integer such that a is not divisible by p (i.e., a and p are co prime), then:

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

- This can be restated in a slightly different form:

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

Groups

A group is a set G and a binary operation # defined on G such that

- There is an identity e∈G, st. e#g = g for g∈G
- Every g∈G has an inverse h∈G st. h#g=e.
- **Associativity** – for all f,g,h∈G, f#(h#g) = (f#g)#h
- **Commutativity** – for all g,h∈G, g#h = h#g.

Order of G = No. of elements in G

How do we prove that?

Lets see what a 'Group' is first...

- The set of integers **{1,2,3,...,p−1} modulo** p forms a group under multiplication. This set is denoted $Zp*$

Some of its properties –

- This group has p−1 elements.
- The operation is multiplication modulo p.
- Every element in $Zp*$ has an inverse in the group, i.e., for every a in $Zp*$, there exists some b such that $a \cdot b \equiv 1 (\mathrm{mod} p)$

Notice, that none of the set elements is 0. And all p-1 elements will be distinct.

So, equating products mod p, we get

$$a^{(p-1)} * (p-1)! \equiv (p-1)! \ (\mathrm{mod} p)$$

So, **a^(p-1)≡1(modp)**

Factoring

Why is factoring key to cryptography?

- imagine you had to multiply 16903 and 32059
- and in the other case factorize 541893277.

Why is it hard? Dont I know that 50% of numbers will have a factor 2, 33.33% have a factor 3 and so on...?

Cyclic Group

Let G be a finite group of order m (written multiplicatively).
Let g be in G
Consider the set { g^0, g^1, g^2, ... }
Also, g^m = g^0 = 1, so that the set has at max m elements.

If in this way, we get a set of m elements, we say **'g' is a generator for the set G.**

Examples

1) **Additive Group Z_n**
- Cyclic for any N, 1 is always a generator, G = {0,1,....,N-1}

2) **Additive Group Z_8**
- Take g = 3,
- we get the set {0, 3, 6, 1, 4, 7, 2, 5}
- how do we get this? {3*0, 3*1, 3*2, ..., 3*7} all taken mod 8.
- The set we get has 8 elements, so g = 3 is a generator.

Examples

3) **Multiplicative Group Z_11**

- Take g = 2,
- we get the set {1, 2, 4, 8, 5, 10, 9, 7, 3, 6}
- how do we get this? {2^0, 2^1, 2^2, ..., 2^10} all taken mod 11.
- The set we get has 11 disctinct elements, so g = 2 is a generator for the set G

Lets get hands on now!

Question 1

Prove that any prime number $p \equiv 1 \bmod 4$ can be expressed as the sum of two squares.

Question 1

Prove that any prime number $p \equiv 1 \bmod 4$ can be expressed as the sum of two squares.

Key points to note –

- **Every square is either 0(mod4) or 1(mod4)**
  - (2n)^2 = 4n^2 = 0(mod4)
  - (2n+1)^2 = 4n^2 + 4n + 1 = 1(mod4)
- Then what can be the sum of 2 possible squares –
  - 0 + 0 = 0(mod4) – Not a prime
  - 1 + 1 = 2(mod4) = 0(mod2) – Not a prime
  - 0 + 1 = 1(mod4) – Can be a prime

Question 1

Prove that any prime number $p \equiv 1 \bmod 4$ can be expressed as the sum of two squares.

The exact proof is complicated but if you wanna know how its done exactly –

See this link

Question 2

Solve the system of congruences:

$$x \equiv 2 \pmod{3}$$
$$x \equiv 3 \pmod{5}$$
$$x \equiv 2 \pmod{7}$$

$$x \equiv 2 \pmod{3}$$
$$x \equiv 3 \pmod{5}$$
$$x \equiv 2 \pmod{7}$$

This is a typical **Chinese Remainder Theorem** question

- So let, x = 2mod3, ie. x = 3k+2
- Now, 3k+2 = 3mod5 => So, 3k = 1 mod 5
- So, k = 2 mod 5 => k = 5m+2
- Then, x = 3k+2 = 15m+8
- So, 15m+8 = 2 mod 7 => 15m = 1 mod7 => m = 1 mod 7 => m = 7n+1
- So, x = 15(7n+1)+8 = 105n+23
- **Hence, x = 23 mod 105**

What next?

We'll see how the number theory concepts we saw today (and some others) help in some very important concepts in cryptography and generation of keys (public and private) !

THANK YOU :)