Cryptography

Ojasw Upadhyay, Matt Steele

CS 2051 — November 3, 2021

§1 Recap

Definition (Fermat's Primality Test)**.** The input is $n \in \mathbb{N}$. The output is Y/N representing whether n is prime?

- Pick* $a_i \in 2, 3, ..., n-1 \mid \gcd(a, n) = 1, 1 \leq i \leq k$
- If $a^{n-1} \equiv 1 \mod n$, `return Y`
- Else: `return N`

Note 1.1 (Carmichael Numbers) — What is the probability of "n is not prime AND Fermat's test outputs Y?"

- $S_1 = (a : 2 \leq a \leq n-1 \mid a^{n-1} \not\equiv 1(\mod n))$, good choices
- $S_2 = (a : 2 \leq a \leq n-1 \mid a^{n-1} \equiv (\mod n))$, bad choices. $a * a^{n-2}$ is the inverse for a^{n-1}

Lemma 1.2

$$|S_2| < |S_1|$$

If we have $S_2 = a_1, a_2, \ldots, a_k$ and $b \in S_1$ such that we have $a_1 b, a_2 b, \ldots, a_k b$. We can state that $a_i b \in S_1$ because

$$(a_i b)^{n-1} = a_i^{n-1} b^{n-1} \equiv 1 \cdot b^{n-1} \not\equiv 1 \mod n.$$

Note. Assume $a_i b \equiv a_j b \mod n$. Multiply by the inverse of a_i, $b \equiv a_i^- 1 * a^j * b \mod n$. $a_i b \in S_1$ because

$$(a_i b)^{n-1} = a_i^{n-1} * b^{n-1} \equiv 1 * b^{n-1} \not\equiv 1 \mod m.$$

Additionally, we have $a_i b \not\equiv a_j b$ because since the $\gcd(b, n) = 1$, the inverse of $b \mod n$ exists! Multiplying both sides by $b^- 1$ we get $a_i \not\equiv a_j(\mod n)$. □

Definition (Fermat's Test on Non-primes)**.** The probability that $\boxed{n}$ is classified correctly is $> \left(\frac{1}{2}\right)^k$. Since we have $n = 10111..01101$, which has length N, the amount of successes if you do something multiple times increases as we get 50/50 probability over time.

§2 Cryptography

We are connecting the powers of reduction ($\mod n$) with the existence of inverses and the divisors of 0, "The coolest application of it."

We have $x \in \mathbb{N}$ where the message is encoded by a number x. Maybe your enemies have the code, and you have to encode that guy to still hide it even if they have the code, you send the next number $n^e \mod N$. The recovery part of the message, the decoding of the message, is when we move the message to $(x^e)^d \equiv x \mod N$. Since we know that $x^{\phi(n)+1} \equiv x \mod N$ (as long as $g.c.d(x, N) = 1$), we can use this to recover our message by taking the message to the power of the Euler totient of the mod. "We don't really need to do Euler's function we can do periodicity of the exponents to see the pattern that lays out the congruence classes."

We need $e \cdot d \equiv 1 \mod \phi(N)$

The key is the pair (N, e) where $x < N$. "Everyone knows the key". Everyone should be able to compute this. In principle you should be able to compute d if you have everything else

Example (Prime N) — Take N prime.
$e \cdot d \equiv 1 \mod N - 1$ is not a good choice!

§2.1 Encoding and Decoding with Primes P and Q

If we give a number that is very large and we are told to compute Euler's function what do we do?

Remember $\phi(N) = p_1^{a_1-1} \cdot p_2^{a_2-1} \cdot p_3^{a_3-1} \cdots p_k^{a_k-1}$. If we have enough primes, we can safe because the enemy will not be able to decode it. Take $N = pq$ where p, q are primes. We tell the world the product of this number. Finding the two numbers from this large number would be extremely difficult. In this case $\phi(N) = (p-1)(q-1)$. Note we need the $\gcd(e, \phi(N)) = 1$. We want that $x^e >> N$. If you are the only one with p and q, you are the only one who can decode the message.

Example — We are given the values of N and e as 55 and 3, respectively. The $\phi(55) = (11-1)(5-1) = 40$. Take $3^3 = 9 \mod 55$, take message $9^27 \equiv 3 \mod 40$. 3 is the original message and this is what we getting. The rest of the world needs to find the prime numbers p and q before they can begin decoding the message.

Example (Another example) **—** We have $N = 7*13 = 91$, $\phi(91) = (7-1)(13-1) = 72$, and $e = 5$ as we need $\gcd(e, \phi(N)) = 1$.

Given $(N, e) = (91, 5)$ and $x = 4$, we send $4^5 \mod 91$ and $4^5 \equiv 23 \mod 91$. To recover the message, we need an inverse of $5 \mod 72$ where 5 is e, 72 is the output of Euler's function of n. The inverse of $5 \mod 72 = 29$

We compute $23^29(mod91) \equiv 4$.

We have 2 messages x and y and we want to check that for the process we go back to the right numbers. We get that $(N+4)^e = y^e$. By sending $x^e \equiv (N+4)^e mod(N)$, we can break the message into small pieces to make sure that the size of x doesn't break the equivalence. For example, for $X = 3N + 4$, we can split X into smaller segments such as $X = (< N) + (< N) + (< N) + (< N)$. If we have $x > N$, one needs to break message as needed so each part is $< N$.

Now lets say we have 2 numbers both $< N$, $x^e \equiv y^e(modN)$ where $x, y < N$ and $g.c.d(x, N) = g.c.d(y, N) = 1$,

$$x^e \equiv y^e \mod N$$
$$\left(x^{-1}\right)^e x^e \equiv (x^{-1})^e y^e$$
$$1 \equiv (x^{-1}y)^e$$

§2.2 Addendum from Textbook: Caesar Ciphers

Caesar's encryption method can be represented by the function f that assigns to the nonnegative integer p, $p \leq 25$, the integer $f(p)$ in the set $0, 1, 2, \ldots, 25$ with

$$f(p) = (p + n) \mod 26.$$

To recover the original message, the function f^{-1}, the inverse of f, is used. Since the cipher is a *shift cipher*, we can set

$$f^{-1}(p) = (p - n) \mod 26$$

where n is the key which is integral to the security of the cipher.