Massachusetts Institute of Technology
6.857: Network and Computer Security
Professors Ronald L. Rivest and Yael Tauman Kalai

Handout R1
February 7, 2020
Adrian Sy

Recitation 1: Math Review

1 Modular Arithmetic

Modular Arithmetic is a system of arithmetic within a finite set of integers, where numbers "wrap around" when reaching a certain number. For example,

$$2 \equiv 12 \equiv 22 \mod 10$$

Generally, for $n > 0$ and integers a and b, we can say that $a \equiv b \mod n$ if n divides $a - b$, also denoted as $n|a - b$.

2 Basic Operations

For addition, subtraction and multiplication, modular arithmetic operations are consistent with normal math. So for a, b, a', b' where $a \equiv a' \mod n, b \equiv b' \mod n$, we have the following:

- $a + b \equiv a' + b' \mod n$
- $a - b \equiv a' - b' \mod n$
- $a \cdot b \equiv a' \cdot b' \mod n$

Division is trickier and relies on the existence of multiplicative inverses. Given an integer a, a^{-1} is the multiplicative inverse of a modulo n if $a \cdot a^{-1} \equiv 1 \mod n$.

Note that a only has a multiplicative inverse modulo n if and only if $\gcd(a, n) = 1$. An outline of the proof is as follows:

1. If $\gcd(a, n) = 1$, then using the extended Euclidean algorithm we can find x, y such that $ax + ny = \gcd(a, n) = 1$. Since $n|(ax - 1)$ then $ax \equiv 1 \mod n$, so x is the multiplicative inverse of a.
   (The algorithm, which also lets us calculate the explicit multiplicative inverse, is described here: `https://en.wikipedia.org/wiki/Extended_Euclidean_algorithm`.)

2. If $\gcd(a, n) \neq 1$, then letting $d = \gcd(a, n)$, we can see that for any integers b, c, $d|ab$ and $d|n$ so regardless of choice of b, $d|(ab - cn)$. But if a had a multiplicative inverse then $d|1 - cn$ by setting b to be a^{-1} but this implies $d|1$ which is impossible for $d \neq 1$.

On the other hand because for any non-zero element in modulo p where p is prime, there exists a multiplicative inverse so division works all the time. Other than the method of computing the multiplicative inverse, division in modular arithmetic works the same as normal division.

For example, $\frac{1}{2} + \frac{1}{3} = \frac{5}{6}$ over rationals. Considering modulo 7, we get $2^{-1} = 4, 3^{-1} = 5, 6^{-1} = 6$ so applying this to the fractional equation $1 \cdot 4 + 1 \cdot 5 \equiv 9 \equiv 5 \cdot 6 \mod 7$.

3 Modular Exponentiation

Exponentiation works the same way as it does in normal math, i.e. repeated multiplication. The interesting part about exponentiation is how the value cycle as we go through the exponents.

For example, the powers of 2 appear as follows when simplified modulo 7.

2^1	2^2	2^3	2^4	2^5	2^6
2	4	1	2	4	1

In this case, it cycles every 3 elements. For any a and prime p, $a^1, a^2, \ldots, a^{p-1}$ must cycle through some subset of the $p-1$ possible residues modulo p. The length of the cycle is also called the order, i.e. $ord_7(2) = 3$.

For any prime p, there exists some integer g such that $ord_p(g) = p - 1$, in other words, the cycle loops through all possible residues. g is often referred to as a primitive root or generator of p.

4 Cryptographic Applications

Several classes of cryptographic schemes rely on g^k being easy to compute ($\log(k)$ operations by using repeated squaring) and finding k from g^k being hard (known as the "discrete logarithm" assumption).

For example, the Diffie-Hellman key exchange described below protocol relies on this assumption to securely create a shared public key between two parties over a public channel.

1. Alice and Bob agree on a large prime p and a generator g for p.
2. Alice randomly generates a number a, Bob randomly generates a number b.
3. Alice sends g^a to Bob, Bob sends g^b to Alice.
4. Alice and Bob independently calculate g^{ab} and use this is a shared key.

Extra: OTP basics

One-time pad (OTP) is an encryption technique that is information theoretically secure but requires a shared key by both parties that is at least as long as the message being encrypted.

Let m be the bit-string representing the message and p be the bit-string representing the pad. For simplicity, assume they are the same length. The encryption scheme is as follows where $\oplus$ is the bit-wise XOR operation:

- To encrypt, generate ciphertext $c = m \oplus p$
- To decrypt, recover message $m = c \oplus p$

For encryption schemes, the usual requirements are correctness and security. Correctness simply means that $Dec_k(Enc_k(m)) = m$, i.e. decryption should give the same message as the encrypted message. Security has several varying definitions which can be either stronger or weaker.

In the case of OTP, correctness is satisfied because $m = (m \oplus p) \oplus p = m \oplus (p \oplus p) = m$. It also has perfect secrecy in the sense that for any ciphertext c and any potential message m' there is a possible pad $p' = c \oplus m'$; as such, seeing only c does not tell us anything about the message because all pads are equally likely. Full proof will be covered in lecture.