CS-303 : Notes

Charlie Stuart : src322

Winter 2021

Section headers with (M-X.X) refer to Mobius modules sections X.X

Contents

1 Course Calendar

What material was covered in what assignments?
Assignment 1 : Index of Coincidence, Vigenere Cypher
Assignment 2 : Modular Matrices, Ring Properties, Modular Domains, RSA Tracing and Implementation, Chrem and RSA
Assignment 3 : Square Roots Mod P, Tonelli-Shanks Implementation, Chrem and Modular Square Roots, Goldwasser Micali Implementation
Assignment 4 : Blum Coin Flipping Protocol
Lab 1 : Learn Maple
Lab 2 : Diophantine Equations, GCD, EEA, Modular Arithmetic
Lab 3 : Modular Inverses, Modular Arithmetic, Affine Cyphers
Lab 4 : GCD, Euler Phi
Lab 5 : Modular Numbers, Euler Phi
Lab 6 : Chinese Remainder Theorem, Fast Powering
Lab 7 : Psuedoprimes, Rabin-Miller Primality,
Lab 8 : Primitive Roots, DLP, Jacobi/Legendre Symbols,
Lab 9 : Birthday Collision, Shanks Babystep-Giant Step
Lab 10 : ElGamal Implementation, Blum Coin Flipping Analysis
Quiz 1P : Logs and Lists in Maple
Quiz 1R : GCD Recursive, Number Theory Proof
Quiz 2P : Chinese Remainder Problem
Quiz 2R : Euler Phi, Breaking RSA
Quiz 3P : RSA Man-In-The-Middle Attacks, RSA Signatures, Rabin-Miller Analysis, Goldwasser-Micali Analysis, Quadratic Residues
Quiz 3R : Quadratic Residues and Symmetric, ElGamal Analysis
Quiz 4P : Tonelli-Shanks Analysis, Quadratic Residues, DLP
Quiz 4R : Factoring Analysis, Blum Coin Flipping Protocol

2 Functions

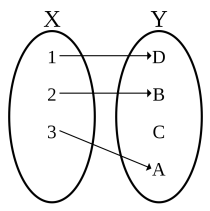
$\mathbb{N}$: Set of natural numbers
$\mathbb{Z}$: Set of integers
$\mathbb{Q}$: Set of rational numbers
$\mathbb{A}$: Set of algebraic numbers
$\mathbb{R}$: Set of real numbers

Domain : Input to a function
Range : Output of a function

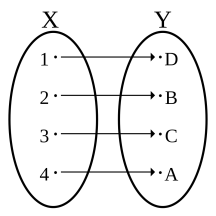
1-1 : Every item in the domain produces a unique result
Onto : Every item in the range of the function can be produced by some element in the domain

Injection : A function that is 1-1, not necessarily onto
Bijection : A function that is 1-1 and onto
Surjection : A function that is onto, not necessarily 1-1

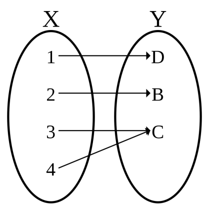
Injection



Bijection



Surjection



Summation : Uses $\sum$, the addition of multiple elements
Pi Notation : Uses $\prod$, like summation but with multiplication

3 Fast Powering (M-3.2)

A lot in crypto the operation $a^p \mod m$ is computed. For secrity, a^p can be massive and hard to compute. Fast powering is a recursive algorithm that reduces the number and complexity of the operations. The following is the fast powering algorithm/function

$$f = \begin{cases} (x^{n/2})^2 & n \text{ is even} \\ x * (x^{(n-1)/2})^2 & n \text{ is odd} \end{cases}$$

This just breaks it down into many $x^2 * x^2 *$ etc etc etc
In Maple:

```
FP := proc(x, b)
        if b = 0 then: return 1:
        else if b = 1 then: return x:
        else if b = 2 then: return x*x:
        else if x mod 2 = 0 then:
                return FP(x, b/2) ^ 2:
        else if x mod 2 = 1 then:
                return FP(x, b/2) ^ 2 * x:
        end if:
end proc:
```

4 Modular Arithmetic (M-3.2)

Some arithmetic

$$\begin{aligned}(x*y) \mod z &= (x \mod z)*(y \mod z)\\ (x+y) \mod z &= ((x \mod z)+(y \mod z)) \mod z\end{aligned}$$

Fermat's Theorem: Given a prime number p and and a where $gcd(a,p) = 1$ then $a^{p-1} = 1 \mod p$ and $a^p = a \mod p$ for all a and also a^{p-2} is the multiplicative inverse of a in $\mathbb{Z}_p$

4.1 Divisibility (M-2.2)

(You know this, it's just new symbols that are confusing you)
b is divisible by a when $b\%a = 0$ this can be written as $a \mid b$
For any integer m, $1 \mid m$ and $m \mid m$
$a \equiv b \mod n$ means $n \mid (b-a)$ which also means there is a q where $a = qn + b$
Properties :

- **Reflexive:** $a \equiv a \mod n$
- **Symmetric:** $(a \equiv b \mod n) \implies (b \equiv a \mod n)$
- **Transitive:** $((a \equiv b \mod n)$ and $(b \equiv c \mod n)) \implies (a \equiv c \mod n)$

4.2 Equivalence Classes

The groups that a mod breaks a set of numbers into. So 5 has five equivalence classes, 0 mod 5, 1 mod 5, 2 mod 5. 3 mod 5, and 4 mod 5. They are the following.

First	0 mod 5	$\{-10,-5,0,5,10\}$
Second	1 mod 5	$\{-9,-4,1,6,11\}$
Third	2 mod 5	$\{-8,-3,2,7,12\}$
Fourth	3 mod 5	$\{-7,-2,3,8,13\}$
Fifth	4 mod 5	$\{-6,-1,4,9,14\}$

The notation is as following

$$[a] = \{x : x = a \mod n\} = \{a + qn \mid q \in \mathbb{Z}\}$$

Then. going back to the arithmetic above

$$\begin{aligned}[][a]+[b] &= [a+b]\\ [a]*[b] &= [ab]\end{aligned}$$

Since mods are sets, commonly we use $a \equiv b \mod m$ which is true iff m is divisible by $a - b$

4.3 Representation

Two forms of representation:
Positive Representation : Only positive values $0 ... m-1$. In Maple, use *mod* or $modp(i, m)$
Symmetric Representation : Includes negative values $m/2 ... m/2$ (kinda). In Maple, use $mods(i, m)$
Both have the same number of values in their representation, and both are circular.
When $m = 5$

i	modp	mods
-5	0	0
-4	1	1
-3	2	2
-2	3	-2
-1	4	-1
0	0	0
1	1	1
2	2	2
3	3	-2
4	4	-1
5	0	0

4.4 Euler Phi Function

Phi is also called the totient function. $\phi(n)$ is number the values between 1 and n that are coprime with n.
If n is prime, then $\phi(n) = n - 1$
Some fun properties:

$$\text{Given } a \text{ and } b \text{ are relatively prime: } \phi(a * b) = \phi(a) * \phi(b) \tag{1}$$

4.4.1 Euler's Theorem

Given an a and m where $gcd(a, m) = 1$, then the following is true:

$$\begin{aligned} a^{\phi(m)} &= 1 \mod m \\ a * a^{\phi(m)-1} &= 1 \mod m \\ a^{\phi(m)-1} &\in a^{-1} \end{aligned}$$

Since $a^{\phi(m)-1} \mod m$ can be computed through fast powering, we can use it as the representation for multiplicative inverses

4.5 Inverses

The modular inverse is also called the multiplicative inverse.
In the domain of $\mathbb{Z}_m$, the multiplicative inverse of a is b where $a * b = 1$.
Not all elements in $\mathbb{Z}_m$ have inverses. a has an inverse in $\mathbb{Z}_m$ iff $gcd(a, m) = 1$
If m is prime, then all non-zero elements have inverses. If m is not prime, then $\phi(n)$ elements have inverses in $\mathbb{Z}_m$
If integers a and b are relatively prime, then a has an inverse in $\mathbb{Z}_b$

4.6 Matrices

So we're going back to linear algebra. So quick, given the following matrix, we can do a bunch of different things with it.

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

Determinant : A special number for square matrices. Used in finding the inverse in our context here. Given the above 2x2 matrix, the determinant is calculated by $ad - bc$
Modular : We can make this modular now by doing the following:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \mod m$$

The new determinant is $ad - bc \mod m$.
Identity Matrix : A matrix where all values are 0 except for the top left to bottom right diagonal, which are 1s.
Inverses : The inverse of a matrix A is when $A * A^{-1}$ creates the identity matrix. The inverse can be found even when taking the modulus of a matrix.
Given that the determinant of matrix $A \mod m$ is d and d and m are coprime (gcd of 1), then an $A^{-1} \mod m$ exists where $A * A^{-1} \mod m$ creates an identity matrix.

4.7 Rings (M-5.1)

Ring : A set of elements R where $0, 1 \in R$ and has the following properties
Additive Identity : $\forall x \in \mathbb{Z}_m, 0 + x = x$
Multiplicative Identity : $\forall x \in \mathbb{Z}_m, 1 * x = x$
Additive Inverse : x and y are additive inverses if $x, y \in \mathbb{Z}_m, x + y = 0$

- There is an addition like operation that is commutative ($a + b = b + a$)
- There is a multiplication like operation that is associative ($ab * c = a * bc$)
- Is distributive over addition ($ab + ac = a(b + c)$)
- There is a multiplicative identity element ($ab = ba = 1$)

$\mathbb{Z}$ is a ring which means $\mathbb{Z}_x$ where x is an integer is a ring

$$x + y \in \mathbb{Z}_4$$

+	a	b	c	d
a	a	b	c	d
b	b	c	d	a
c	c	d	a	b
d	d	a	b	c

+	0	1	2	3
0	0	1	2	3
1	1	2	3	0
2	2	3	0	1
3	3	0	1	2

$$x * y \in \mathbb{Z}_4$$

*	a	b	c	d
a	a	a	a	a
b	a	b	c	d
c	a	c	b	c
d	a	d	c	b

*	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	1	2
3	0	3	2	1

4.7.1 Fields (M-7.1)

Fields : Rings where all non-zero elements are units
Unit : A element with an inverse
Finite Field : A field with a finite set of elements
$\mathbb{F}$ is a field when:

- $\mathbb{F}$ has a set of elements where operations + and * are defined and $\mathbb{F}$ is closed under those operations
- $\mathbb{F}$ has an additive identity and multiplicative identity (often written as 0 and 1). Distributive law still holds: $\forall a, b, c \in \mathbb{F}, a(b + c) = ab + ac$
- $\mathbb{F}$ is a commutative ring
- Quotient and remainder work in $\mathbb{F}$
- All non-zero elements have a unit $\forall x \neq 0 \in \mathbb{F} \exists y \in \mathbb{F}$ such that $xy = 1$

$\mathbb{Z}_5$ is a field. $\mathbb{Z}_6$ is not a field. If p is prime, then $\mathbb{Z}_p$ is a field.

5 Psuedoprime numbers

Psuedoprimes, also called probably primes, are numbers that is not actually prime, but shares properties with prime numbers. Classified based on which prime numbers they share properties with.

5.1 Fermat Psuedoprimes

(see Fermats theorem above). Given Fermats theorem, given prime p and coprime a, then $a^{p-1} - 1$ is divisible by p. For an integer $a > 1$, if a not prime (composite) integer x, and $a^{x-1} - 1$ is divisible by x, then x is psuedoprime to the base a.

5.2 Miler-Rabin Primality Test

It tests whether a given number is likely to be prime or not. These are called strong probable primes. This one has a lot of letters, don't get too confused.

Given a number n that is an odd integer > 2, then $n = 2^s * d + 1$, where s and d are both positive, and d is odd.

Considering a base a where $0 < a < n$, then n is a strong probable prime to a if one of the two following hold.

$$a^d \equiv 1 \mod n$$
$$a^{2^r * d} \equiv -1 \mod n \text{ where } 0 \leq r < s$$

This is both because of Fermats Theorem, and because the only square roots of $1 \mod n$ are 1 and -1.

The Miller-Rabin Primality Test is 75% accurate in finding if a number is composite, and will fail to find if a number is composite 25% of the time.

5.3 Prime Number Theorem

Let $\pi(X)$ be the number of primes p where $2 \leq p \leq X$ then

$$\lim_{X \to \infty} \left(\frac{\pi(X)}{\frac{X}{\ln(X)}}\right) = 1$$

In not math human terms, as X approaches infinity, the number of primes between 2 and X over the natural log of X is 1, which means they're basically the same. I don't understand what the natural log means.

Wikipedia says, for large enough X, the probability that a random integer not greater than X is prime is very close to $1/\log(X)$

So basically, this limit tells me, if I grab an integer less than X, how likely is that number to be prime.

6 PreModern Ciphers (M-1.1, 1.2, 2.1)

6.1 Caesar Cipher

Shift cipher
All that's needed to break is the shift number. Can be found finding the most frequent letter and substituting with e

Alphabet	A	B	C	...	X	Y	Z
Encypted	F	G	H	...	C	D	E

6.1.1 Attacking

Since it is known that the shift cipher is being used, all that is needed is the shift value which is between 0 and $n-1$ where n is the length of the alphabet.
Chosen Plaintext : After getting "special messages" $m1$, $m2$, $m3$, and $m4$, encrypt "A" (compute σ("A")) which finds the shift for all letters. $1+n=c \mod 26$
Chosen Cyphertext : Decrypt "A" (compute σ^{-1}("A")) $m+n=1 \mod 26$
Cyphertest Only : Given encrypted message $c1$, $c2$, and $c3$ figure out σ and σ^{-1} and the keys being used.
Brute Force : Test all n since n is small

6.1.2 Affine Cyphers

An affine cypher is a generalization of a ceaser cypher. In an alphabet of n elements and a key k, then $E(m)=m+k \mod n$
Given an a where $gcd(a,n)=1$ (a and n are coprime), then an affine cypher with the key (a,k) encrypts using the following function: $E(m)=am+k \mod n$

6.2 Substitution Cipher

Alphabet	A	B	C	...	X	Y	Z
Encypted	Z	Q	A	...	R	T	Y

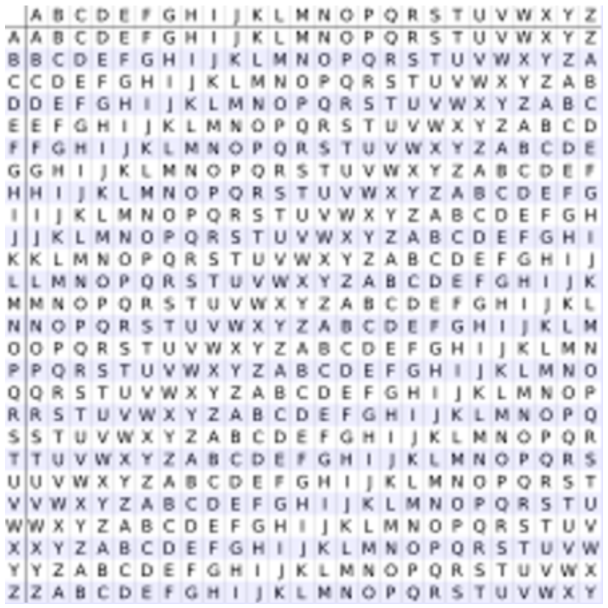
Permutation of letters
Needs the full map to break. Given an alphabet of n characters, there are $n!$ possible combinations

6.3 One Time Pad

Easy, we know this from CS164. Given a message m and a key k, then $E(m)=m$ **XOR** k and $D(m)=E(m)$ **XOR** k

6.4 Vigenere Cypher

A polyalphabetic shift where, with each letter I send, I use a different shift. No long brute forcable. $n!$ possible different shifts for an alphabet of n characters.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Given the above key table of shifts where the column is a letter from the key code and row is the plain text letter, the key code of “GAY”, and the plain text phrase, “IAMCHARLIE” is encrypted to be “OAKIHYXLGK”. See the table below for a breakdown. As you can see, “I” was encoded as both “O” and “G” depending on the shift. Frequency analysis is impossible now.

Plain Text	**Key Letter**	**Result**
I	G	O
A	A	A
M	Y	K
C	G	I
H	A	H

While not a Vigenere cypher, a polyalphabetic substitution cypher is possible where there is a sequence of substitutions. Now, given an alphabet of n characters and a plaintext message of length m, there are $n!^{length(m)}$ possible sequences of substitutions. The key in this situation assigns each substitution group an integer between 1 and 26!, then that string of integers is applied to each respective plain text character.

7 Euclidian Algorithm (M-3.1)

An algorithm for finding the greatest common divisors of two integers a and b. It works by computing a remainder sequence as follows

$$\begin{aligned} r_0 &\leftarrow a \\ r_1 &\leftarrow b \\ r_2 &\leftarrow r_0 \% r_1 \\ r_3 &\leftarrow r_1 \% r_2 \\ &\ldots\ldots \\ r_{n-1} &\leftarrow gcd(a,b) \\ r_n &\leftarrow 0 \end{aligned}$$

7.1 Diophantine Equations

Diophantine Equation : An equation where all the knowns and unknowns are integers

Given a modulus m and a number $a \in \mathbb{Z}_m$, we want to find a t such that $at \equiv 1 \mod m$. We could brute force, but that takes too long. We know that t is the inverse of a, so to find this inverse, we need to find a u and v where $au + mv = 1$

Since there are two unknowns, one equation, and only integers allowed, there may be 0 or many solutions to any equation. $4u + 2v = 1$ has no solutions while $5u + 2v = 1$ has many.

7.2 Extended Euclidian Algorithm

The Extended Euclidian Algorithm (EEA) computes answers to the previous diophantine equations. In addition to the previous remainder sequence, other sequences are produced.

$\boldsymbol{i}$	$\boldsymbol{q}$	$\boldsymbol{r}$	$\boldsymbol{s}$	$\boldsymbol{t}$
0		$r_0 \leftarrow a$	$s_0 \leftarrow 1$	$t_0 \leftarrow 0$
1	$q_1 \leftarrow r_0/r_1$	$r_1 \leftarrow b$	$s_1 \leftarrow 0$	$t_1 \leftarrow 1$
2	$q_2 \leftarrow r_1/r_2$	$r_2 \leftarrow r_0 - q_1 r_1$	$s_2 \leftarrow s_0 - q_1 s_1$	$t_2 \leftarrow t_0 - q_1 t_1$
...	...	...	...	...
i	$q_i \leftarrow r_{i-1}/r_i$	$r_i \leftarrow r_{i-2} - q_{i-1} r_{i-1}$	$s_i \leftarrow s_{i-2} - q_{i-1} s_{i-1}$	$t_i \leftarrow t_{i-2} - q_{i-1} t_{i-1}$
...	...	...	...	...
$n-1$		$r_{n-1} \leftarrow gcd(a,b)$	$s_{n-1} \leftarrow u$	$t_{n-1} \leftarrow v$
n		$r_n \leftarrow 0$		

This process is stopped when $r_n = 0$. In this case, the results r_{n-1}, s_{n-1}, and t_{n-1} are relevant as they are the solutions as follows:

$$\begin{aligned} a(s_{n-1}) + b(t_{n-1}) &= r_{n-1} \\ au + bv &= gcd(a, b) \end{aligned}$$

Now applying this to what we're doing with modular arithmetic. Suppose we have some $a < m$ where $gcd(a, m) = 1$. Given that u and v are solutions to $au + vm = 1$, then $u \equiv a^{-1} \mod m$ which means that u is the inverse of a in $\mathbb{Z}_m$.

8 Chinese Remainder Theorem (M-5.1)

Given moduli m and n where $gcd(m, n) = 1$, and residues a and b, there exists an integer x that solves:

$$\begin{aligned} x &\equiv a \mod m \\ x &\equiv b \mod n \end{aligned}$$

8.1 Proof

Consider the map $x \to (x \mod m, x \mod n)$
This map is a 1-1 map from $\mathbb{Z}_{mn}$ to $\mathbb{Z}_m \times \mathbb{Z}_n$, since if x and y map to the same pair, then $x \equiv y \mod m$ and $x \equiv y \mod n$. Since $gcd(m, n) = 1$, this implies that $x \equiv y \mod mn$
Since there are mn elements in both $\mathbb{Z}_{mn}$ and $\mathbb{Z}_m \times \mathbb{Z}_n$, the map is also onto. This means every pair (a, b) we can find the desired x
Another Way:
Let $\mathbb{Z}_m \times \mathbb{Z}_n$ denote the set of pairs (a, b) where $a \in \mathbb{Z}_m$ and $b \in \mathbb{Z}_n$. We can perform the following arithmetic on $\mathbb{Z}_m \times \mathbb{Z}_n$ by performing componentwise modular arithmetic.

$$\begin{aligned} (a, b) + (c, d) &= (a + b, c + d) \\ (a, b)(c, d) &= (ac, bd) \end{aligned}$$

From there, we can get:

$$\begin{aligned} (ac \mod m, bd \mod n) &= (a \mod m, b \mod n)(c \mod m, d \mod n) \\ (a + c \mod m, b + d \mod n) &= (a \mod m, b \mod n) + (c \mod m, d \mod n) \end{aligned}$$

For every pair (a, b) there is an integer x where $x \mod m, x \mod n) = (a, b)$

8.2 Constructive

If $gcd(m, n) = 1$ there exists e_m and e_n such that:

$$\begin{aligned} e_m &\equiv 1 \mod m \\ e_m &\equiv 0 \mod n \\ e_n &\equiv 0 \mod m \\ e_n &\equiv 1 \mod n \end{aligned}$$

This follows that $ae_m + be_n \equiv a \mod m \equiv b \mod n$
Since $gcd(m, n) = 1$, the EEA says there's an x and y for $my + ny = 1$. Just set $e_m = ny$ and $e_n = mx$

8.3 Solving Two Congruences

So given the following example:

$$\begin{aligned} x &\equiv 2 \mod 3 \\ x &\equiv 3 \mod 8 \end{aligned}$$

We do the following to solve this congruence:

$$\begin{aligned}
x &\equiv 3 \mod 8 && \\
x &= 8j + 3 && \text{Convert congruence into equivalent equation} \\
8j + 3 &\equiv 2 \mod 3 && \text{Set that equation equal to the other congruence} \\
j &\equiv 1 \mod 3 && \text{Solve for } j \\
j &= 3k + 1 && \text{Convert congruence into equivalent equation} \\
x &= 8(3k + 1) + 3 && \text{Substitute } j \text{ in } x \text{ equation} \\
x &= 24k + 11 && \text{Solve} \\
x &\equiv 11 \mod 24 && \text{Convert to congruence}
\end{aligned}$$

This final congruence is the combination of the two given, it's solution solves both the others.

8.4 Solving With a Summation

While we didn't learn this in class, I think it's pretty great so I'm gonna talk about it
Just a summation to solve for x given many moduli and residues. Given the following where all r are integers and all m are coprime and positive integers:

$$\begin{aligned} x &\equiv r_1 \mod m_1 \\ x &\equiv r_2 \mod m_2 \\ &\ldots\ldots \\ x &\equiv r_n \mod m_n \end{aligned}$$

To solve this, we compute the following:

$M = m_1 * m_2 * ... * m_n$	The product of all moduli
$M_i = M/m_i$	M divided by the current moduli
$s_i = M_i^{-1} \mod m_i$	Using EEA, the inverse of M_i in m_i

Then we can plug this into the following summation:

$$x = \sum_{i=1}^{n} r_i M_i s_i$$

From here, x may need to be modded by M.

9 RSA Encryption (M-4.2)

The general process is as follows.

1. Choose random large prime numbers p and q. These are secret
2. Calculate $n = pq$
3. Calculate $\phi(n) = (p-1)(q-1)$
4. **Public Key** : Choose an integer e such that $1 < e < \phi(n)$ and $gcd(e, \phi(n)) = 1$
5. **Private Key** : Choose an integer d such that $de \equiv 1 \mod \phi(n)$

Public : n, e
Private : p, q, d
Every person has their own set of n, e, and d. For Alice to send a message to Bob, Alice encrypts her message with n_b and e_b so that only Bob can decrypt it with his d_b.
This is then encrypted by doing $c = M^{e_b} \mod n_b$ so that Bob can compute $M = c^{d_b} \mod n_b$

9.1 Breaking with Chinese Remainder (M-5.1)

Given a plain text message M, if we can prove that $M^{ed} \mod p = M \mod p$ and $M^{ed} \mod q = M \mod q$, then we can use the CRT on these two equations.
Given the following known information:

$n = pq$	RSA Calculation
$\phi(n) = (p-1)(q-1)$	Euler Phi product rule
$ed \equiv 1 \mod \phi(n)$	RSA Calculation
$ed = k\phi(n) + 1$	Modular equation
$a^p \equiv a \mod p$	Little Fermat

We can do the following process to prove $M^{ed} \mod p = M \mod p$:

$M^{ed} \mod p$	
$M^{k\phi(n)+1} \mod p$	Using modular equivalent equation
$M^{k\phi(n)} * M \mod p$	Exponent arithmetic
$M^{k(p-1)(q-1)} * M \mod p$	Euler Phi product rule
$M^{(p-1)^{k(q-1)}} * M \mod p$	Exponent arithmetic
$1^{k(q-1)} * M \mod p$	Little Fermats
$M \mod p$	1 to any power is 1

Now CRT can be applied to the following equations to find $M^{ed} \mod pq = M$

$$\begin{aligned} M^{ed} \mod p &= M \mod p \\ M^{ed} \mod q &= M \mod q \end{aligned}$$

9.2 Signatures (M-5.2)

Signatures are a unique identifier to verify the sender of a message. Digital signatures are extra information in a message to verify the sender.
So let's say Alice wants to send another message to Bob, but in a way so he can verify it's from her. She will do the following:

$$\begin{aligned} M_s &= M^{d_a} \mod n_a && \text{Encrypt with her private info} \\ M_c &= M_s^{e_b} \mod n_b && \text{Encrypt with Bob's public info} \end{aligned}$$

So that Bob can do the following:

$$\begin{aligned} M_s &= M_c^{d_b} \mod n_b && \text{Decrypt with his private info} \\ M &= M_s^{e_a} \mod n_a && \text{Decrypt with Alice's public info} \end{aligned}$$

Assuming there were no attacks:

- Anyone can decrypt Alice's signature, since e_a and n_a are public and used to decrypt
- Only Alice can encrypt her signature, since d_a is private and used to encrypt
- Anyone can send Bob an encrypted message, since e_b and n_b are public and used to encrypt
- Only Bob can decrypt messages sent to him, since d_b is private and used to decrypt

So first encrypting the messages with Alices signature, which only she can encrypt, then encrypting the message with Bobs information (which anyone can encrypt). Bob then can be the only person to decrypt the message, then can ensure that only Alice could have sent her signature.

9.3 Man-In-The-Middle Attacks (M-6.1)

When Alice is encrypting a message to send to Bob, there is a person in the middle, Eve, who is catching the messages before they reach Bob, decrypting them, then sending either the same message, or new messages to Bob.
The process (simplified) can go as follows:

1. Alice sends Bob a message m_1, Eve intecepts
2. Eve sends a message to Alice "Please give me your public key info".
3. Alice, assuming she is talking to Bob, sends her n_a and e_a.
4. Eve sends the same message to Bob to get his n_b and e_b.
5. Eve sends them both their n_e and e_e pretending it is Alice/Bobs respective public keys
6. Eve now can intercept the messages, and also send new messages to Alice and Bob without either of them being suspicious

10 Post-Midterm Math

10.1 Primitive Roots (M-7.1)

Given a positive integer p, a is a primitive root mod p if for every integer relatively prime to p, there is a power of a that is congruent.
Root of Unity : a is a root of unity if there exists a k where $a^k \equiv 1 \mod p$
See the following table for $\mathbb{F}_5^*$. 4 is a root of unity. 2 and 3 are primitive roots.

a^i	a^0	a^1	a^2	a^3	...
1	1	1	1	1	:(
2	1	2	4	3	:)
3	1	3	4	2	:)
4	1	4	1	4	:(

10.2 Quadratic Residues (M-7.1)

p is an odd prime number, a is any number where p doesn't divide a so $a \mod p \neq 0$ a is a quadratic residue mod p if a is a square mod p. This means $\exists x \in \mathbb{Z}_p x^2 \equiv a \mod p$
Now, given what we know about symmetric and positive representation, we can assume the following (in $\mathbb{Z}_7$). We know 6 and -1 are additive inverses, along with -2 and 5, and -3 and 4. Along with this, we know x^2 is always positive no matter whether x is positive or negative. This means if we want to compute all the square roots of numbers mod 7, we only need to compute them for 1, 2, and 3, as the results will hold. See the table below. 0 was omitted as it has no additive inverse. I'm using a^{-1} as additive inverse in the below table.

a	1	2	3	4	5	6
a^{-1}	-6	-5	-4	-3	-2	-1
$a^2 \mod p$	1	4	2	2	4	1

Extending the above regarding $(-x)^2 = x^2$, we can derive the following:

$$\begin{aligned}(-x)^2 &= (p-x)^2\\ (p-x)^2 &= p^2 - 2xp + x^2\\ p^2 - 2xp + x^2 &\equiv a \mod p\end{aligned}$$

Now some properties (given p is an odd prime):

- The product of two quadratic residues mod p is also a quadratic residue mod p
- The product of a quadratic residue mod p and a quadratic nonresidue mod p is not a quadratic residue mod p
- The product of two quadratic nonresidues mod p is a quadratic residue mod p

10.3 Euler's Criterion

Now Euler, as always a much smarter man than I am, found that a is a quadratic residue where an x exists such that $a \equiv x^2 \mod p$, p is prime, and a is coprime to p iff:

$$a^{\frac{p-1}{2}} \equiv 1 \mod p$$

In addition to this, b is a quadratic nonresidue iff:

$$b^{\frac{p-1}{2}} \equiv -1 \mod p$$

10.3.1 Legendre Symbols

Given p is an odd prime, the Legendre Symbol of a is:

$$\left(\frac{a}{p}\right) = \begin{cases} -1 & a \text{ is a quadratic nonresidue mod } p \\ 0 & p \mid a \\ 1 & a \text{ is a quadratic residue mod } p \end{cases}$$

Now we love fun properties!!!

- **Product Rule** : $\left(\frac{a}{p}\right) * \left(\frac{b}{p}\right) = \left(\frac{ab}{p}\right)$
- **Reduction Rule** : $a \equiv b \mod p \implies \left(\frac{a}{p}\right) = \left(\frac{b}{p}\right)$
- **Quadratic Reciprocity** : Given p and q are prime, then: $\left(\frac{p}{q}\right) = \left(\frac{q}{p}\right)$

10.3.2 Jacobi Symbol

Jacobi symbol is good for avoiding knowing the factors of a numerator, or when the numerator is prime so that a reduction is still possible.
Given a and b integers where b is odd and positive, and a is anything. b's prime factorization is:

$$b = p_1^{e_1} * p_2^{e_2} * ... * p_n^{e_n}$$

The resulting Jacobi symbol is:

$$\left(\frac{a}{b}\right) = \left(\frac{a}{p_1}\right)^{e_1} * \left(\frac{a}{p_2}\right)^{e_2} * ... * \left(\frac{a}{p_n}\right)^{e_n}$$

More properties because we can't get enough!

- $\left(\frac{0}{a}\right) = 0$
- $\left(\frac{1}{a}\right) = 1$
- **Product Rule** : For "numerators" $\left(\frac{ac}{b}\right) = \left(\frac{a}{b}\right) * \left(\frac{c}{b}\right)$
- **Product Rule** : For "denominators" $\left(\frac{a}{bc}\right) = \left(\frac{a}{b}\right) * \left(\frac{a}{c}\right)$
- **Reduction Rule** : $\left(\frac{a}{b}\right) = \left(\frac{a \mod b}{b}\right)$
- **Quadratic Reciprocity** : If a and b are both odd, then $\left(\frac{a}{b}\right) = \left(\frac{b}{a}\right)$ If $a \equiv 3 \mod 4$ **AND** $b \equiv 3 \mod 4$ then $\left(\frac{a}{b}\right) = -\left(\frac{b}{a}\right)$

Proof Regarding mod 4
This mod 4 thing is really confusing to me, and I'm still confused why we choose the number 4, buy anyway. Given that p is a prime number, p mod 4 will NEVER equal 2 or 0 since that would require p being even, which it's not and will never be. Given the base case $a = -1$ we can do the following:

$$\begin{aligned}
p &\equiv 1 \mod 4 \\
p &= 4k+1 \\
\left(\frac{a}{p}\right) &\equiv a^{\frac{p-1}{2}} \mod p \\
\left(\frac{-1}{p}\right) &\equiv -1^{\frac{(4k+1)-1}{2}} \\
\left(\frac{-1}{p}\right) &\equiv -1^{\frac{4k}{2}} \\
\left(\frac{-1}{p}\right) &\equiv -1^{2k} \\
\left(\frac{-1}{p}\right) &\equiv 1
\end{aligned}
\qquad\qquad
\begin{aligned}
p &\equiv 3 \mod 4 \\
p &= 4k+3 \\
\left(\frac{a}{p}\right) &\equiv a^{\frac{p-1}{2}} \mod p \\
\left(\frac{-1}{p}\right) &\equiv -1^{\frac{(4k+3)-1}{2}} \\
\left(\frac{-1}{p}\right) &\equiv -1^{\frac{4k+2}{2}} \\
\left(\frac{-1}{p}\right) &\equiv -1^{2k+1} \\
\left(\frac{-1}{p}\right) &\equiv -1
\end{aligned}$$

10.3.3 Finding Quadratic Roots

While Legendre/Jacobi symbol answers where a number $a \mod p$ has a square root, it does not give a way to compute the square root.
When $p \equiv 3 \mod 4$ and a is a quadratic residue mod p, then $x^2 = a^{\frac{p+1}{4}} \mod p$

10.3.4 Tonelli-Shanks Algorithm

Used to find x when $x^2 \equiv a \mod p$ when p is prime. This is helpful when $p \not\equiv 3 \mod 4$
Given that p is prime and a is a quadratic residue of p.
The Implementation In Maple:

```
TS := proc(a, p)
        # to find Q and S where p-1 = Q2^S
        Q := p - 1:
        S := 0:
        while Q mod 2 = 0 do:
                S := S + 1:
                Q := Q / 2:
        end do:

        # find a quadratic nonresidue z
        z := 1:
        while z^((p-1)/2) mod p = 1 do:
                z := z + 1;
        end do:

        # Set up
        M := S mod p:
        c := z^Q mod p:
        t := a^Q mod p:
        R := a^((Q+1)/2) mod p:

        # Loop
        while 1 = 1 do:
                if t = 0 then:
                        return 0;
                elif t = 1 then:
                        return R;
                else:
                        i := 0;
                        while i < M do:
                                if t^(2^i) mod p = 1 then:
                                        break:
                                end if:
                                i := i + 1:
```

```
                end do:

                b := c^(2^(M-i-1)) mod p:
                M := i:
                c := (b * b) mod p:
                t := (t * b * b) mod p:
                R := (R * b) mod p:
            end if;
        end do:
end proc:
```

10.4 Birthday Paradox

This is regarding the probability of collisions kinda relating the to pidgeonhole principle.
Pidgeonhole Principle : If there are n holes and $n+1$ elements there will be a collision when trying to sort them
There are two main questions to answer

1. What is the probability someone has the same birthday as you?
2. What is the probability at least two people share the same birthday?

The answer to question 1 is simple, the chance someone has the same birthday as me out of n people is the same as one minus the chance someone doesnt have the same birthday as me. That means for 1 person not having the same birthday as me, theres a 364/365 chance. With n people:

$$P(n) = 1 - \prod_{i=1}^{n} \frac{364}{365}$$
$$P(n) = 1 - (\frac{364}{365})^n$$

Now for question 2, we want the chances of no one having the same birthday, then find the inverse of that. So now we see the following:

$$P(n) = 1 - \prod_{i=1}^{n} \frac{365-(i-1)}{365}$$

This sets us up for finding the probability of collisions in Discrete Logarithm Problems

11 Goldwasser-Micali Encryption (M-7.2)

The first probabilistic public-key encryption scheme which is provably secure under standard cryptographic assumptions. Not efficient, the cyphertexts may be several hundred times longer than the original plain text message. There are three main parts/subalgorithms to this algorithm

Public Key : (N, a) where $N = pq$ and x is a quadratic residue mod N. Can be verified knowing the factorization of N

Private Key : (p, q), the factorization of N

11.1 Part 1 : Key Generation

1. Generate distinct large primes p and q randomly and independently
2. Compute $N = pq$
3. Find an x such that $\left(\frac{x}{p}\right) = \left(\frac{x}{q}\right) = -1$ This x value can be found by selecting random values and testing. If $p \equiv 3 \mod 4$ and $q \equiv 3 \mod 4$ then $N - 1$ is guaranteed to work

11.2 Part 2 : Message Encryption

1. Encode m as a string of bits $m_1, ..., m_n$
2. For every m_i, generate a random y_i where $gcd(y_i, N) = 1$
3. $c_i = y_i^2 x^{m_i} \mod N$

11.3 Part 3 : Message Decryption

1. For each i, using determine if c_i is a quadratic residue, if so $m_i = 0$ otherwise, $m_i = 1$
2. m = $m_1, ..., m_n$

12 Discrete Logarithm Problems (M-7.2, 8.2)

Given a finite field $\mathbb{F}_p^*$ and a primitive root g of that field, a discrete log problem is: given a value h, solve $g^x = h \mod p$ for x. Can also be written as $\log_g(h) = x$

12.1 Brute Force

Test values of x until a valid one is found. This runs in $O(x)$ time where x is an unknown value.

12.2 Shanks Algorithm

Las Vegas Algorithm : A randomized algorithm that always gives the correct answer but may use a lot or a little resources
Monte Carlo Algorithm : A randomized algorithm who's output may be incorrect with certain, typically small probability. Use repeated random sampling to obtain numeric results

I don't know which Shanks is yet.

Once again, given our DLP where $g^x = hmodp$, there needs to be an easier way than brute forcing to find the problem. We first get our variables.

$$\begin{aligned} N &= p - 1 \\ n &= 1 + \lfloor\sqrt{N}\rfloor \end{aligned}$$

From there, two lists are created:

$$\begin{aligned} l1 &= 1, g^1, g^2, ..., g^n \\ l2 &= h, hg^{-n}, hg^{-2n}, ... \end{aligned}$$

Use these two lists to find a match where:

$$\begin{aligned} l1_i &= l2_j \\ g^i &= hg^{-jn} \\ h &= g^i g^{jn} \end{aligned}$$

Using simple algebra, we then find that the solution $x = i + jn$

13 ElGamal Encryption (M-8.1)

13.1 Key Generation

1. Pick a large prime p
2. Choose a g that is high order
3. Pick an $a > 0$ such that $A = g^a \mod p$

Public Key : A, g, p
Private Key : a
Given large enough g and p the DLP is super difficult to solve.

13.2 Encryption

Given an encoded message m where $0 < m < p$:

1. Pick a random number $0 < k < p$
2. Compute $c_1 = g^k \mod p$
3. Compute $c_2 \equiv mA^k \mod p$

13.3 Decryption

Given a cyphertext (c_1, c_2):

1. Compute $x \equiv (c_1^a)^{-1} \mod p$ using EEA
2. Compute $x * c_2 \equiv m$

The justification:

$$\begin{aligned}
x * c_2 &\equiv (c_1^a)^{-1} * c_2 && \text{Definition of } x \\
x * c_2 &\equiv (g^{ak})^{-1} * c_2 && \text{Definition of } c_1 \\
x * c_2 &\equiv (g^{ak})^{-1} * (m * A^k) && \text{Definition of } c_2 \\
x * c_2 &\equiv (g^{ak})^{-1} * (m * (g^a)^k) && \text{Definition of } A \\
x * c_2 &\equiv (g^{ak})^{-1} * g^{ak} * m \\
x * c_2 &\equiv 1 * m \\
x * c_2 &\equiv m \mod p
\end{aligned}$$

13.4 With Digital Signatures

ElGamal with digital signatures does not encrypt the plain text message. The message is used to perform calculations, but it is not encrypted. The signature is computed with:

$$\begin{aligned} S_1 &= g^k \mod p \\ S_2 &= (D - aS_1)k^{-1} \mod (p-1) \end{aligned}$$

Then verified with:

$$(A^{S_1} S_1^{S_2} \mod p) = (g^D \mod p)$$

14 Blum Algorithm (M-9.1)

Also called the Blum-Micali Algorithm. A cryptographically secure psuedorandom number generator. The security comes from the difficulty of computing discrete logarithms.
Given p is an odd prime and g it's primitive root and a seed x_0, then

$$x_{i+1} = g^{x_i} \mod p$$

The ith output is 1 if $x_i \leq \frac{p-1}{2}$, otherwise it's 0.

14.1 Blum Coin Flipping Protocol

Knowing that digital logarithm problems are very hard to solve, we can use that concept to create a secure remote coin flipping protocol.

1. A and B agree to a large prime p such that $p \mod$, and a primitive root $a \in \mathbb{F}_p^*$
2. A will "flip the coin" and B will "call heads/tails"
3. A chooses a random x
4. A computes $y = a^x \mod p$
5. A sends y to B
6. B guesses even or odd for the value of x
7. A sends B the value of x
8. B computes a^x is y and whether their call was correct or not

Since DLPs are very hard to solve, there is no way for A to lie and choose a specific y to indicate the parity of x and theres no way, looking at y for B to guess the parity of x.

14.2 Coin Flipping with RSA

Assume all messages are sent via RSA with both A and B encrypting each message with a signature.

1. B picks an $n = pq$ where p and q are large odd primes and $p \equiv q \equiv 3 \mod 4$. This is separate from his RSA N
2. n has a publication date P. An 8 char string ddmmyyyy. This is only valid in the past 5 years.
3. B sends A n and P
4. A checks $n \equiv 1 \mod 4$
5. A picks a random $x \in \mathbb{Z}_n$

6. A computes $y = x^2 \mod n$
7. A sends n, P, y
8. B verifies n and P
9. B picks b +1 or -1 for heads and tails respectively
10. B sense n, P, y, and b
11. A verifies n, P, y
12. A sends x
13. B wins if $b = \left(\frac{x}{n}\right)$

15 Quantum Computing

15.1 Factoring

Small numbers with lots of small factors are easy to factor.

15.1.1 Fermats Algorithm

15.1.2 Pollard p-1

15.1.3 Quadratic Sieve

15.1.4 Shor's

15.2 Qubits