5 Number Theory

5.1 Introduction

Division
Let a and b be integers. We say that a divides b, or $a|b$ if:

$$\exists d \text{ s.t. } b = ad$$

If $b \neq 0$ then $|a| \leq |b|$.
Division Theorem: For any integer a and any positive integer n, there are unique integers q and r such that $0 \leq r < n$ and $a = qn + r$.
The value $r = a \bmod n$ is called the remainder or the residue of the division.
Theorem: If $d|a$ and $d|b$ then $d|(xa+yb)$ for any integers x, y.
Proof: $a = rd$ and $b = sd$ for some r, s. Therefore, $xa + yb = xrd + ysd = d(xr + ys)$, so $d|(xa+yb)$

Greatest Common Divisor
For integers a and b:
The greatest common divisor gcd(a,b) is defined as follows:

$$\gcd(a,b) = \max(d \text{ s.t. } d|a \text{ and } d|b) \ (a \neq 0 \text{ or } b \neq 0).$$

Note: This definition satisfies gcd(0,1) = 1.
The lowest common multiple lcm(a,b) is defined as follows:

$$\text{lcm}(a,b) = \min(m > 0 \text{ s.t. } a|m \text{ and } b|m) \ (\text{for } a \neq 0 \text{ and } b \neq 0).$$

a and b are coprimes (or relatively prime) iff gcd(a,b) = 1.

Prime Numbers
An integer $p \geq 2$ is called prime if it is divisible only by 1 and itself.
Fundamental Theorem of Arithmetic: every positive number can be represented as a product of primes in a unique way, up to a permutation of the order of primes.
There are infinitely many primes

- Euclid gave simple proof by contradiction (c. 300BC).

The number of primes $\leq n : \pi(n) \approx n/\ln n$

- Even though the number of primes is infinite, the density of primes gets increasingly sparse as $n \to \infty$.

5.2 Modular Arithmetic

Modular Arithmetic
Modular arithmetic is fundamental to modern public key cryptosystems.
Given integers $a, b, n \in \mathbb{Z}$ we say that a is congruent to b modulo n:

$$a \equiv b \pmod{n} \text{ iff } n \text{ divides } b - a$$

Often we are lazy and just write $a \equiv b$ if it is clear we are working modulo n.
The modulo operator is like the C-operator %.
Example: $16 \equiv 1 \pmod{5}$ since 16-1 = 3 $\times$ 5

Modular Arithmetic
For convenience we define the set:

$$\mathbb{Z}_n = \{0, \ldots, n-1\}$$

which is the set of remainders modulo n.
It is clear that given n, every integer $a \in \mathbb{Z}$ is congruent modulo n to an element in the set $\mathbb{Z}_n$, since we can write:

$$a = q \times n + r$$

with $0 \leq r < n$ and $a \equiv r \pmod{n}$

Modular Arithmetic
The set $\mathbb{Z}_n$ has two operations defined on it.

- Addition
  - $11 + 13 \mod 16 \equiv 24 \pmod{16} \equiv 8 \pmod{16}$.
- Multiplication
  - $11 \times 13 \pmod{16} \equiv 143 \pmod{16} \equiv 15 \pmod{16}$.

Given integers $a, b \in \mathbb{Z}$ we have:

- $a + b \pmod{n} \equiv (a \pmod{n} + b \pmod{n}) \pmod{n}$
- $a - b \pmod{n} \equiv (a \pmod{n} - b \pmod{n}) \pmod{n}$
- $a \times b \pmod{n} \equiv (a \pmod{n} \times b \pmod{n}) \pmod{n}$

Modular Arithmetic
Properties of modular addition:

- Closure:
$$\forall a, b \in \mathbb{Z}_n : a + b \in \mathbb{Z}_n$$
- Associativity:
$$\forall a, b, c \in \mathbb{Z}_n : (a + b) + c \equiv a + (b + c)$$
- Commutativity:
$$\forall a, b \in \mathbb{Z}_n : a + b \equiv b + a$$
- Identity (0 is the identity):
$$\forall a \in \mathbb{Z}_n : a + 0 \equiv 0 + a \equiv a$$
- Inverse ($n - a$ is the inverse of a):
$$\forall a \in \mathbb{Z}_n : a + (n - a) \equiv (n - a) + a \equiv 0$$

Modular Arithmetic
Properties of modular multiplication:

- Closure:
$$\forall a, b \in \mathbb{Z}_n : a \times b \in \mathbb{Z}_n$$
- Associativity:
$$\forall a, b, c \in \mathbb{Z}_n : (a \times b) \times c \equiv a \times (b \times c)$$
- Commutativity:
$$\forall a, b \in \mathbb{Z}_n : a \times b \equiv b \times a$$
- Distributivity (distrubutes over addition):
$$\forall a, b, c \in \mathbb{Z}_n : (a + b) \times c \equiv a \times c + b \times c$$
- Identity (1 is the identity) :
$$\forall a \in \mathbb{Z}_n : a \times 1 \equiv 1 \times a \equiv a$$

Multiplicative Inverse
Division a/b in modular arithmetic is performed by multiplying a by the multiplicative inverse of b.
The multiplicative inverse of $b \in \mathbb{Z}_n$ is an element denoted $b^{-1} \in \mathbb{Z}_n$ with:
$$bb^{-1} \equiv b^{-1}b \equiv 1$$

Theorem: $b \in \mathbb{Z}_n$ has a unique inverse modulo n iff b and n are relatively prime i.e. gcd(b,n) = 1.
Theorem: If p is a prime then every non-zero element in $\mathbb{Z}_p$ has an inverse.

Multiplicative Inverse
Consider $\mathbb{Z}_{10}$:

- 3 has a multiplicative inverse, since gcd(3,10)=1.
  - $3 \times 7 \equiv 21 \equiv 1 \pmod{10}$.
- 5 has no multiplicative inverse, since gcd(5,10)=5.
  - We have the following table:

$$\begin{array}{ll} 0 \times 5 \equiv 0 \pmod{10} & 5 \times 5 \equiv 5 \pmod{10} \\ 1 \times 5 \equiv 5 \pmod{10} & 6 \times 5 \equiv 0 \pmod{10} \\ 2 \times 5 \equiv 0 \pmod{10} & 7 \times 5 \equiv 5 \pmod{10} \\ 3 \times 5 \equiv 5 \pmod{10} & 8 \times 5 \equiv 0 \pmod{10} \\ 5 \times 5 \equiv 0 \pmod{10} & 9 \times 5 \equiv 5 \pmod{10} \end{array}$$

Greatest Common Divisor (GCD)
We need a method to determine when $a \in \mathbb{Z}_n$ has a multiplicative inverse and compute it when it does.
We know this happens iff gcd(a,n) = 1.
Therefore we need to compute the GCD of two integers $a, b \in \mathbb{Z}$.

- This is easy if we know the prime factorization of a and b, since:

$$a = \prod p_i^{\alpha_i} \text{ and } b = \prod p_i^{\beta_i} \Rightarrow d = \gcd(a,b) = \prod p_i^{min(\alpha_i,\beta_i)}$$

- However, factoring is a very expensive operation, so we cannot use the above formula.
- A much faster algorithm to compute GCDs is Euclid's algorithm.

GCD - Euclidean Algorithm
To compute the GCD of a and b we compute:

$$\begin{array}{lcl} a & = & q_0 b + r_0 \\ b & = & q_1 r_0 + r_1 \\ & \vdots & \\ r_{k-2} & = & q_{k-1} r_{k-1} + r_k \\ r_{k-1} & = & q_k r_k \end{array}$$

If d divides a and b then d divides r_0, r_1, r_2 and so on.
Therefore: $\gcd(a, b) = \gcd(b, r_0) = \gcd(r_0, r_1) = \cdots = \gcd(r_{k-1}, r_k) = r_k$

GCD - Euclidean Algorithm
As an example of this algorithm we want to show that:

$$3 = \gcd(21,12)$$

Using the Euclidean algorithm we compute gcd(21,12) as:

$$\begin{aligned}
\gcd(21,12) &= \gcd(21 \pmod{12},12) \\
&= \gcd(9,12) \\
&= \gcd(12 \pmod{9},9) \\
&= \gcd(3, 9) \\
&= \gcd(9 \pmod{3},3) \\
&= \gcd(0,3) = 3
\end{aligned}$$

XGCD - Extended Euclidean Algorithm
Using the Euclidean algorithm, we can determine when a has an inverse modulo n i.e. iff $\gcd(a,n) = 1$.

- But we do not know yet how to compute the inverse.

Solution: use an extended version of the Euclidean algorithm.
Recall that during the Euclidean algorithm we had:

$$r_{i-2} = q_{i-1}r_{i-1} + r_i$$

and finally $r_k = \gcd(a,b)$.
Now we unwind the above and write each r_i in terms of a and b.

XGCD - Extended Euclidean Algorithm
Unwinding the various steps in the Euclidean algorithm gives:

$$\begin{aligned}
r_0 &= a - q_0 b \\
r_1 &= b - q_1 r_0 = b - q_1(a - q_0 b) = -q_1 a + (1 + q_0 q_1) b \\
. &\quad \vdots \\
r_k &= xa + yb
\end{aligned}$$

The XGCD takes as input a and b and outputs x, y, r_k such that:

$$r_k = \gcd(a,b) = xa + yb$$

XGCD - Multiplicative Inverse
Given $a, n \in \mathbb{Z}$ we can compute d, x, y using XGCD such that:

$$d = \gcd(a,n) = xa + yn$$

Considering the above equation modulo n we get:

$$d \equiv xa + yn \pmod{n} \equiv xa \pmod{n}$$

Thus if $d = 1$ then a has a multiplicative inverse given by:

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

Remark: the more general equation $ax \equiv b \pmod{n}$ has precisely $d = \gcd(a,n)$ solutions iff d divides b.

Modular Exponentiation

Given a prime p and $a \in \mathbb{Z}_n^*$ we want to calculate $a^x \pmod{n}$.
It does not make sense to compute $y = a^x$ and then $y \pmod{n}$.
Consider $123^5 \pmod{511} = 28153056843 \pmod{511} = 359$
There is a large intermediate result so this method takes a very long time and a great deal of space for large a, x and n.
$123^5 \pmod{511}$ could also be calculated as follows:

$$\begin{array}{lcl} a & = & 123 \\ a^2 & = & a \times a \pmod{511} = 310 \\ a^3 & = & a \times a^2 \pmod{511} = 316 \\ a^4 & = & a \times a^3 \pmod{511} = 32 \\ a^5 & = & a \times a^4 \pmod{511} = 359 \end{array}$$

This requires four modular multiplications; it is still far too slow.

Modular Exponentiation

It is much better to compute this example using the steps below:

$$\begin{array}{lcl} a & = & 123 \\ a^2 & = & a \times a \pmod{511} = 310 \\ a^4 & = & a^2 \times a^2 = 310 \times 310 \pmod{511} = 32 \\ a^5 & = & a \times a^4 = 123 \times 32 \pmod{511} = 359 \end{array}$$

This requires only 3 multiplications.
This shows that if we consider the binary representation of the exponent $x = x_{k-1}x_{k-2}\ldots x_1x_0$, then the value represented by each bit of the exponent x_i can be obtained by squaring the value represented by the previous bit x_{i-1}.
Multiplication is required for every bit which is set after the first one.
Thus for an exponent with k bits of which j bits are set, $k-1$ squarings and $j-1$ multiplications.

Modular Exponentiation

This suggests an algorithm which works through the exponent one bit at a time squaring and multiplying.
This is commonly known as the square and multiply algorithm.
Right to left variant for calculating $y = a^x \pmod{n}$:

```
y = 1
for i = 0 to k-1 do
   if x_i = 1 then y = y*a (mod n)
   a = a*a (mod n)
end
```

Left to right variant for calculating $y = a^x \pmod{n}$:

```
y = 1
for i = k-1 downto 0 do
   y = y*y (mod n)
   if x_i = 1 then y = y*a (mod n)
end
```

Chinese Remainder Theorem (CRT)
Consider $n = 15 = 3 \times 5$.
We can represent every element a of $\mathbb{Z}_n$ by its coordinates $(a \pmod{3}, a \pmod{5})$.
This leads to the following table:

	0	1	2	3	4
0	0	6	12	3	9
1	10	1	7	13	4
2	5	11	2	8	14

Note that all elements in $\mathbb{Z}_n$ have different coordinates, i.e. given (a_1,a_2) with $0 \leq a_1 < 3$ and $0 \leq a_2 < 5$ we can reconstruct a.

Chinese Remainder Theorem (CRT)
Consider $n = 24 = 4 \times 6$.
We can represent every element a of $\mathbb{Z}_n$ by its coordinates $(a \pmod{4}, a \pmod{6})$.
This leads to the following table:

	0	1	2	3	4	5
0	0/12		8/20		4/16	
1		1/13		9/21		5/17
2	6/18		2/14		10/22	
3		7/19		3/15		11/23

Note that a and $a + 12 \pmod{24}$ map to the same coordinates.
Therefore, given (a_1,a_2) with $0 \leq a_1 < 4$ and $0 \leq a_2 < 6$ we cannot uniquely reconstruct a.

Chinese Remainder Theorem (CRT)
The previous examples indicate that if $n = n_1 \times n_2$ with $\gcd(n_1,n_2) = 1$, we can replace computing modulo n by computing modulo n_1 and modulo n_2:

$$\mathbb{Z}_n \cong \mathbb{Z}_{n_1} \times \mathbb{Z}_{n_2} \text{ iff } \gcd(n_1,n_2) = 1$$

If $n = n_1 \times n_2$ then it is very easy to compute the coordinates of $a \in \mathbb{Z}_n$, since these are simply $(a \pmod{n_1}), a \pmod{n_2})$.
However, given the coordinates (a_1, a_2) with $0 \leq a_1 < n_1$ and $0 \leq a_2 < n_2$ how do we compute the corresponding a?

Chinese Remainder Theorem (CRT)
We can reformulate our reconstruction problem as:
Given: $n = n_1 \times n_2$ with $\gcd(n_1, n_2) = 1$
Compute: $x \in \mathbb{Z}_n$ with $x \equiv a_1 \pmod{n_1}$ and $x \equiv a_2 \pmod{n_2}$
Example: If $x \equiv 4 \pmod 7$ and $x \equiv 3 \pmod 5$ then we have:

$$x \equiv 18 \pmod{35}$$

How did we work this out?

CRT - Example
We want to find $x \in \mathbb{Z}_n$ with $n = 35$ such that:

$$x \equiv 4 \quad \pmod 7 \text{ and } x \equiv 3 \quad \pmod 5$$

Therefore, for some $k \in \mathbb{Z}$ we have:

$$x = 4 + 7k \text{ and } x \equiv 3 \quad \pmod 5$$

Substituting the equality in the second equation gives:

$$4 + 7k \equiv 3 \quad \pmod 5$$

Therefore, k is given by the solution of:

$$2k \equiv 7k \equiv 3 - 4 \equiv 4 \quad \pmod 5$$

Hence we can compute k as $k \equiv 4/2 \pmod 5 \equiv 2 \pmod 5$, so:

$$x \equiv 4 + 7k \equiv 4 + 7 \times 2 \equiv 18 \quad \pmod{35}$$

CRT - General Case
Let $n_1, \ldots, n_k$ be pairwise relatively prime and let $a_1, \ldots, a_k$ be integers.
We want to find x modulo $n = n_1 n_2 \cdots n_k$ such that:

$$x \equiv a_i \pmod{n_i} \text{ for all } i$$

The CRT guarantees a unique solution given by:

$$x = \sum_{i=1}^{k} a_i \times N_i \times y_i \quad \pmod n$$

$$N_i = n/n_i \text{ and } y_i = N_i^{-1} \pmod{n_i}$$

Note that $N_i \equiv 0 \pmod{n_j}$ for $j \neq i$ and that $N_i \times y_i \equiv 1 \pmod{n_i}$

CRT - General Case Example
We want to find the unique x modulo $n = 1001 = 7 \times 11 \times 13$ such that:

$$x \equiv 5 \pmod 7 \text{ and } x \equiv 3 \pmod{11} \text{ and } x \equiv 10 \pmod{13}$$

We compute:

$$N_1 = 143, y_1 = 5 \text{ and } N_2 = 91, y_2 = 4 \text{ and } N_3 = 77, y_3 = 12.$$

Then we reconstruct x as:

$$\begin{array}{rcl} x & \equiv & \sum_{i=1}^{k} a_i \times N_i \times y_i \pmod n \\ & \equiv & 5 \times 143 \times 5 + 3 \times 91 \times 4 + 10 \times 77 \times 12 \pmod{1001} \\ & \equiv & 894 \pmod{1001} \end{array}$$

CRT - Modular Exponentiation
Let $n = 55 = 5 \times 11$ and suppose we want to compute $27^{37} \pmod n$.
This can be done in a number of ways:

- Really stupid: using 36 multiplications modulo 55:

$$(((27 \times 27) \pmod n)) \times 27 \pmod n)) \cdots 27 \pmod n)$$

- Less stupid: using 5 squarings and 2 multiplications modulo 55:

$$((27^{2^5} \pmod n)) \times 27^{2^2} \pmod n)) \times 27 \pmod n)$$

- Rather intelligent: using 5 squarings and 2 multiplications modulo 5 and modulo 11 and CRT to combine both results.

Quadratic Residues
An integer q is called a quadratic residue modulo n if there exists an integer x such that:

$$x^2 \equiv q \pmod n$$

Integer x is called the square root of $q \pmod n$.
If no such integer x exists, q is called a quadratic nonresidue modulo n.

Example (n = 11):

x	0	1	2	3	4	5	6	7	8	9	10
$x^2 \pmod{11}$	0	1	4	9	5	3	3	5	9	4	1

There are five quadratic residues modulo 11: 1, 3, 4, 5, and 9.
There are five quadratic non-residues modulo 11: 2, 6, 7, 8, 10.
The trivial case $x^2 = 0$ is generally excluded from the list of quadratic residues.

Quadratic Residues
If p is a prime exactly half of the numbers in $\mathbb{Z}_p^*$ are quadratic residues.
Euler's Criterion: Given odd prime p and $q \in \mathbb{Z}_p^*$:

- q is a quadratic residue iff $q^{(p-1)/2} \equiv 1 \pmod{p}$.
- q is quadratic nonresidue, iff $q^{(p-1)/2} \equiv -1 \pmod{p}$.

A quadratic residue $q \in \mathbb{Z}_p^*$ cannot be a primitive root, since $q^{(p-1)/2} \equiv 1 \pmod{p}$ and the order of a primitive root is $p-1$.

Quadratic Residues Modulo $n = pq$
Let $n = pq$ where p and q are large primes.
If $a \in \mathbb{Z}_n^*$ is a quadratic residue modulo n, then a has exactly four square roots modulo n in $\mathbb{Z}_n^*$.
Therefore exactly a quarter of the numbers in $\mathbb{Z}_n^*$ are quadratic residues modulo n.

Legendre's Symbol
If p is a prime and a is an integer.
Legendre's symbol $\left(\dfrac{a}{p}\right) = \begin{cases} 0, & \text{if } a|p \\ +1, & \text{if } a \text{ is a quadratic residue modulo } p \\ -1, & \text{if } a \text{ is a quadratic non-residue modulo } p \end{cases}$
By Euler's criterion: $\left(\frac{a}{p}\right) = a^{(p-1)/2} \pmod{p}$.

Legendre's Symbol
Properties of Legendre's symbol:

1. $a \equiv b \pmod{p} \Rightarrow \left(\frac{a}{p}\right) = \left(\frac{b}{p}\right)$
2. $\left(\frac{ab}{p}\right) = \left(\frac{a}{p}\right)\left(\frac{b}{p}\right)$
3. $\left(\frac{1}{p}\right) = 1$
4. $\left(\frac{-1}{p}\right) = \begin{cases} 1, & \text{if } p \equiv 1 \pmod{4} \\ -1, & \text{if } p \equiv 3 \pmod{4} \end{cases}$
5. $\left(\frac{2}{p}\right) = (-1)^{(p^2-1)/8}$
6. If p and q are odd primes: $\left(\frac{p}{q}\right) = (-1)^{((p-1)/2)((q-1)/2)}\left(\frac{q}{p}\right)$

Jacobi's Symbol

Jacobi's symbol is a generalization of Legendre's symbol to composite numbers.
If n is odd with prime factorization $n = p_1 \times p_2 \times \ldots \times p_k$ and a is relatively prime to n:
Jacobi's symbol $\left(\frac{a}{n}\right) = \left(\frac{a}{p_1}\right) \times \left(\frac{a}{p_2}\right) \times \ldots \times \left(\frac{a}{p_k}\right)$
$\left(\frac{a}{n}\right) = -1 \Rightarrow a$ is a quadratic non-residue
$\left(\frac{a}{n}\right) = 1 \not\Rightarrow a$ is a quadratic residue

Jacobi's Symbol

Properties of Jacobi's symbol:

1. $a \equiv b \pmod{n} \Rightarrow \left(\frac{a}{n}\right) = \left(\frac{b}{n}\right)$
2. $\left(\frac{ab}{n}\right) = \left(\frac{a}{n}\right)\left(\frac{b}{n}\right)$
3. $\left(\frac{a}{mn}\right) = \left(\frac{a}{m}\right)\left(\frac{a}{n}\right)$
4. $\left(\frac{1}{n}\right) = 1$
5. $\left(\frac{-1}{n}\right) = (-1)^{(n-1)/2}$
6. $\left(\frac{2}{n}\right) = (-1)^{(n^2-1)/8}$
7. If m and n are odd co-primes: $\left(\frac{m}{n}\right) = (-1)^{((m-1)/2)((n-1)/2)}\left(\frac{n}{m}\right)$

Square Roots Modulo Prime $p \equiv 3 \pmod{4}$

If the Legendre symbol is -1, then there is no solution.
If p is a prime and a is a quadratic residue modulo p then:

$$a^{(p-1)/2} \equiv 1 \pmod{p} \text{ (by Euler's criterion).}$$

Multiplying both sides by a:

$$a^{(p+1)/2} \equiv a \pmod{p}$$

Taking the square roots of both sides:

$$\pm a^{(p+1)/4} \equiv \sqrt{a} \pmod{p}$$

If $p \equiv 3 \pmod{4}$, then $(p+1)/4$ is an integer, and this can be used to calculate the square root.

Square Roots Modulo Prime $p \equiv 5 \pmod 8$
If p is a prime and a is a quadratic residue modulo p then:

$$a^{(p-1)/2} \equiv 1 \pmod p \text{ (by Euler's criterion).}$$

Taking the square roots of both sides:

$$a^{(p-1)/4} \equiv \pm 1 \pmod p$$

If $a^{(p-1)/4} \equiv 1 \pmod p$ then:

$$\sqrt{a} = \pm a^{(p+3)/8} \pmod p$$

If $a^{(p-1)/4} \equiv -1 \pmod p$ then:

$$\sqrt{a} = \pm 2a(4a)^{(p-5)/8} \pmod p$$

If $p \equiv 5 \pmod 8$ then $(p+3)/8$ and $(p-5)/8$ are integers.

Square Roots Modulo Prime $p \equiv 1 \pmod 8$
If p is a prime s.t. $p \equiv 1 \pmod 8$ and a is a quadratic residue modulo p the probabilistic Tonelli-Shanks algorithm can be used to calculate $\sqrt{a}$:

Choose a random n until one is found such that $(n/p) = -1$
Let e, q be integers such that q is odd and $p-1 = 2^e q$
$y = n^q \pmod p$
$r = e$
$x = a^{(q-1)/2} \pmod p$
$b = ax^2 \pmod p$
$x = ax \pmod p$
while $b \neq 1 \pmod p$ **do**
  Find the smallest m such that $b^{2^m} = 1 \pmod p$
  $t = y^{2^{r-m-1}} \pmod p$
  $y = t^2 \pmod p$
  $r = m$
  $x = xt \pmod p$
  $b = by \pmod p$
end
return x

Square Roots Modulo $n = pq$
If the Jacobi symbol is -1, then there is no solution.
If a is a quadratic residue and $\sqrt{a} \pmod p = \pm x$ and $\sqrt{a} \pmod q = \pm y$, then we can use the Chinese Remainder Theorem to calculate $\sqrt{a}$.
Example: Compute the square root of 3 modulo 11×13
$\sqrt{3} \pmod{11} = \pm 5$
$\sqrt{3} \pmod{13} = \pm 4$
Using the Chinese Remainder Theorem, we can calculate the four square roots as 82, 126, 17 and 61.

5.3 Group Theory

Groups
A group $(S, \oplus)$ consists of a set S and an operation $\oplus$, satisfying:

- Closure: $\forall a, b \in S : a \oplus b \in S$
- Associativity: $\forall a, b, c \in S : a \oplus (b \oplus c) = (a \oplus b) \oplus c$
- Identity element e: $\exists e \in S : \forall a \in S : a \oplus e = e \oplus a = a$
- Every element has an inverse element:

$$\forall a \in S : \exists a^{-1} \in S : a \oplus a^{-1} = a^{-1} \oplus a = e$$

- The group S is called commutative or Abelian if:

$$\forall a, b \in S : a \oplus b = b \oplus a$$

The order of a group S, denoted by $|S|$, is the number of elements in S. If a group S satisfies $|S| < \infty$ then it is called a finite group.

Groups
$(\mathbb{Z}, +)$, $(\mathbb{R}, +)$, $(\mathbb{Q}, +)$ and $(\mathbb{C}, +)$ are groups.

- the identity is 0, the inverse of x is $-x$

$(\mathbb{R} \setminus \{0\}, \times)$, $(\mathbb{Q} \setminus \{0\}, \times)$ and $(\mathbb{C} \setminus \{0\}, \times)$ are groups.

- the identity is 1, the inverse of x is $1/x$

These are all examples of infinite Abelian groups.
$(\mathbb{Z}_n, +)$ is a finite Abelian group.
Questions:

- Why is $(\mathbb{Z} \setminus \{0\}, \times)$ not a group?
- Why is $(\mathbb{R}, \times)$ not a group?

Rings
A ring $(S, \oplus, \otimes)$ consists of a set S together with two binary operators $\oplus$ and $\otimes$, satisfying:

- Closure of $\oplus$: $\forall a, b \in S : a \oplus b \in S$
- Associativity of $\oplus$: $\forall a, b, c \in S : a \oplus (b \oplus c) = (a \oplus b) \oplus c$
- Commutativity of $\oplus$: $\forall a, b \in S : a \oplus b = b \oplus a$
- Identity for $\oplus$: $\exists 0 \in S : \forall a \in S : a \oplus 0 = 0 \oplus a = a$

- Inverse for $\oplus$: $\forall a \in S : \exists -a \in S : a \oplus -a = -a \oplus a = 0$
- Closure of $\otimes$: $\forall a,b \in S : a \otimes b \in S$
- Associativity of $\otimes$: $\forall a,b,c \in S : a \oplus (b \otimes c) = (a \otimes b) \otimes c$
- Identity for $\otimes$: $\exists 1 \in S : \forall a \in S : a \otimes 1 = 1 \otimes a = a$
- Distributivity of $\otimes$ over $\oplus$: $\forall a,b,c \in S : a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)$ and $(b \oplus c) \otimes a = (b \otimes a) \oplus (c \otimes a)$

Rings
The following are all examples of infinite rings:

- $(\mathbb{Z}, +, \times)$
- $(\mathbb{R}, +, \times)$
- $(\mathbb{Q}, +, \times)$
- $(\mathbb{C}, +, \times)$

The following is an example of a finite ring:

- $(\mathbb{Z}_n, +, \times)$

Fields
A field $(S, \oplus, \otimes)$ consists of a set S together with two binary operators $\oplus$ and $\otimes$, satisfying all the properties of a ring plus the following:

- Commutativity of $\otimes$: $\forall a,b \in S : a \otimes b = b \otimes a$
- Inverse for $\otimes$: $\forall a \neq 0 \in S : \exists a^{-1} \in S : a \otimes a^{-1} = a^{-1} \otimes a = 1$

So $(S, \oplus)$ is an abelian group with identity 0 and $(S \setminus \{0\}, \otimes)$ is an abelian group with identity 1.
$(\mathbb{R}, +, \times)$, $(\mathbb{Q}, +, \times)$ and $(\mathbb{C}, +, \times)$ are all infinite fields.
$(\mathbb{Z}_n^*, +, \times)$ is a finite field.

Finite Fields
A finite field is a field that contains a finite number of elements.
There is exactly one finite field of size (order) p^n where p is a prime (called the characteristic of the field) and n is a positive integer.
If p is a prime $\mathbb{Z}_p$ is the finite field GF(p) (note here that $n = 1$ and so is omitted).
Finite fields are of central importance in coding theory and cryptography.
GF(2^8) is of particular importance as an element can be represented in a single byte.

Euler Groups
We define the set of invertible elements of $\mathbb{Z}_n$ as:

$$\mathbb{Z}_n^* = \{a \in \mathbb{Z}_n : \gcd(a,n) = 1\}$$

The set $\mathbb{Z}_n^*$ is always a group with respect to multiplication and is called an Euler group.
When n is a prime p we have:

$$\mathbb{Z}_p^* = \{1,\ldots,p-1\}$$

Examples:

$$\begin{array}{ll}
\mathbb{Z}_1 = \{0\} & \mathbb{Z}_1^* = \{0\} \\
\mathbb{Z}_2 = \{0,1\} & \mathbb{Z}_2^* = \{1\} \\
\mathbb{Z}_3 = \{0,1,2\} & \mathbb{Z}_3^* = \{1,2\} \\
\mathbb{Z}_4 = \{0,1,2,3\} & \mathbb{Z}_4^* = \{1,3\} \\
\mathbb{Z}_5 = \{0,1,2,3,4\} & \mathbb{Z}_5^* = \{1,2,3,4\}
\end{array}$$

Euler Totient Function $\phi(n)$
Euler's totient function $\phi(n)$ represents the number of elements in $\mathbb{Z}_n^*$:

$$\phi(n) = |\mathbb{Z}_n^*| = |\{a \in \mathbb{Z}_n : \gcd(a,n) = 1\}|$$

$\phi(n)$ is therefore the number of integers in $\mathbb{Z}_n$ which are relatively prime to n.
We know that an element $a \in \mathbb{Z}_n$ has a multiplicative inverse modulo n iff $\gcd(a,n) = 1$.
Therefore, there are precisely $\phi(n)$ invertible elements in $\mathbb{Z}_n$.

Euler Totient Function $\phi(n)$
Given the prime factorization of n:

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

we can compute $\phi(n)$ using the following formula:

$$\phi(n) = \prod_{i=1}^{k} p_i^{e_i-1}(p_i - 1)$$

The most important cases for cryptography are:

- If p is prime then:

$$\phi(p) = p - 1$$

- If p and q are both prime and $p \neq q$ then:

$$\phi(pq) = (p-1)(q-1)$$

Lagrange's Theorem
The order of an element a of a group $(S, \otimes)$ is the smallest positive integer k such that $a^k = 1$.
Lagrange's Theorem:

$$\text{If } S \text{ is a group of size } |S| = n \text{ then } \forall a \in S : a^n = 1$$

Corollary: the order k of an element $a \in S$ divides $n = |S|$, so if $a \in \mathbb{Z}_n^*$ then k divides $\phi(n)$.
Thus if $a \in \mathbb{Z}_n^*$ then $a^{\phi(n)} \equiv 1 \pmod{n}$, since $|\mathbb{Z}_n^*| = \phi(n)$.
This is known as Euler's Theorem.

Euler's Theorem
Euler's theroem allows us to simplify the calculation of modular exponentiation since the following holds:
$a^k \pmod{n} \equiv a^{k \pmod{\phi(n)}} \pmod{n}$
For example, to calculate $101^{108} \pmod{109}$, we can simplify this as follows:

$$\begin{aligned}
& 101^{108} \pmod{109} \\
= \; & 101^{108 \pmod{\phi(109)}} \pmod{109} \\
= \; & 101^{108 \pmod{108}} \pmod{109} \\
= \; & 101^{0} \pmod{109} \\
= \; & 1
\end{aligned}$$

Fermat's Little Theorem
Not to be confused with Fermat's Last Theorem . . .
If p is a prime, Lagrange's Theorem tells us that $a^{p-1} \equiv 1 \pmod{p}$.
If we multiply both sides of this equation by a, we obtain Fermat's Little Theorem:

$$\text{if } p \text{ is a prime then } a^p \equiv a \pmod{p}$$

Fermat's Little Theorem can be used to test whether a number n is probably a prime: this will be the case if $a^{n-1} \equiv 1 \pmod{n}$.

Generators
For $a \in \mathbb{Z}_n^*$ the set $\{a^0, a^1, a^2, a^3, \ldots\}$ is called the group generated by a, denoted $\langle a \rangle$.
The order of $a \in \mathbb{Z}_n^*$ is the size of $\langle a \rangle$, denoted $|\langle a \rangle|$.
Examples for $\mathbb{Z}_7^*$:
$\langle 3 \rangle = \{1, 3, 2, 6, 4, 5\}$, so the order of 3 is 6
$\langle 2 \rangle = \{1, 2, 4\}$, so the order of 2 is 3
$\langle 1 \rangle = \{1\}$, so the order of 1 is 1

Primitive Roots
$a \in \mathbb{Z}_n^*$ is called a primitive root of $\mathbb{Z}_n^*$ if the order of a is $\phi(n)$.
Not all groups possess primitive roots e.g. $\mathbb{Z}_n^*$ where $n = pq$ and p, q are odd primes.
If $\mathbb{Z}_n^*$ possesses a primitive root a, then $\mathbb{Z}_n^*$ is called cyclic.
If a is a primitive root of $\mathbb{Z}_n^*$ and $b \in \mathbb{Z}_n^*$ then $\exists x$ s.t. $a^x \equiv b \pmod{n}$. This x is called the discrete logarithm or index of b modulo n to the base a.
Examples for $\mathbb{Z}_7^*$:
3 is a primitive root: $\{3^0, 3^1, 3^2, 3^3, 3^4, 3^5\} = \{1, 3, 2, 6, 4, 5\} = \mathbb{Z}_7^*$
2 is not a primitive root: $\{2^0, 2^1, 2^2, 2^3, 2^4, 2^5\} = \{1, 2, 4\} \neq \mathbb{Z}_7^*$

Primitive Roots
A primitive root exists in $\mathbb{Z}_n^*$ iff n has a value 2, 4, p^k or $2p^k$ for some odd prime p and integer k.
To determine whether a is a primitive root of $\mathbb{Z}_n^*$, we need to show for all prime factors $p_1, \ldots, p_k$ of $\phi(n)$ that $a^{\phi(n)/p_i} \neq 1 \pmod{n}$ This can be determined using modular exponentiation.
For a prime p the number of primitive roots mod p is $\phi(p-1)$

5.4 Primality Testing

Prime Numbers
The generation of prime numbers is needed for many public key algorithms:

- RSA: Need to find p and q to compute $N = pq$
- ElGamal: Need to find prime modulus p
- Rabin: Need to find p and q to compute $N = pq$

We shall see that testing a number for primality can be done very fast

- Using an algorithm which has a probability of error
- Repeating the algorithm lowers the error probability to any value we require.

Prime Numbers
Before discussing the algorithms we need to look at some basic heuristics concerning prime numbers.
A famous result in mathematics, conjectured by Gauss after extensive calculation in the early 1800's, is:

Prime Number Theorem The number of primes less than X is approximately $\dfrac{X}{\log X}$

This means primes are quite common.
The number of primes less than 2^{512} is about 2^{503}

Prime Numbers
By the Prime Number Theorem if p is a number chosen at random then the probability it is prime is about:

$$\frac{1}{\log p}$$

So a random number p of 512 bits in length will be a prime with probability:

$$\approx \frac{1}{\log p} \approx \frac{1}{355}$$

So on average we need to select 177 odd numbers of size 2^{512} before we find one which is prime.
Hence, it is practical to generate large primes, as long as we can test primality efficiently

Primality Tests
For many cryptographic schemes, we need to generate large primes. This is usually done as follows:

- Select a random large number
- Test whether or not the number is a prime.

Naive approach to primality testing on n:

- Check if any integer from 2 to n-1 (or better: $\sqrt{n}$) divides n.

An improvement:

- Check whether n is divisible by any of the prime numbers $\leq \sqrt{n}$
- Can skip all numbers divisible by each prime number (Sieve of Eratosthenes)

These methods are too slow.

Sieve of Eratosthenes
To find prime numbers less than n:

- List all numbers $2, 3, 4, \ldots, n-1$
- Cross out all numbers with factor of 2, other than 2
- Cross out all numbers with factor of 3, other than 3, and so on
- Numbers that “fall through” sieve are prime

	2	3	~~4~~	5	~~6~~	7	~~8~~	~~9~~	~~10~~
11	~~12~~	13	~~14~~	~~15~~	~~16~~	17	~~18~~	19	~~20~~
~~21~~	~~22~~	23	~~24~~	~~25~~	~~26~~	~~27~~	~~28~~	29	~~30~~

Primality Tests

Two varieties of primality test:

- Probabilistic
  - Identify probable primes with very low probability of being composite (in which case they are called pseudoprimes).
  - Much faster to compute than deterministic tests.
  - Examples:
    - Fermat
    - Solovay-Strassen
    - Miller-Rabin
- Deterministic
  - Identifies definite prime numbers.
  - Examples:
    - Lucas-Lehmer
    - AKS

Fermat Primality Test

Fermat's Little Theorem: if n is prime and $1 \leq a < n$, then:

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

To test if n is prime, a number of random values for a are chosen in the interval $1 < a < n-1$, and checked to see if the following equality holds for each value of a:

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

If n is composite then for a random $a \in \mathbb{Z}_n^*$:

$$\Pr[a^{n-1} \equiv 1 \pmod{n}] \leq 1/2$$

A composite number n is called a Fermat pseudoprime to base a if $a^{n-1} \equiv 1 \pmod{n}$.

Fermat Primality Test

```
Pick random a, 1 < a < n-1
if a^(n-1) (mod n)=1 then
   return PRIME
else
   return COMPOSITE
end
```

This test can be repeated k times to reduce the probability of classifying composites as primes.
If the algorithm outputs COMPOSITE at least once: output COMPOSITE; this will always be correct (a is called a Fermat witness).
If the algorithm outputs PRIME in all k trials: output PRIME (a Fermat pseudoprime); this will be an error with probability $(1/2)^k$.
Some composites always pass Fermat's test, and so are falsely identified as prime: the Carmichael Numbers.

Fermat Primality Test
Carmichael numbers are composite numbers n which fail Fermat's Test for every a not dividing n.

- Hence probable primes which are not primes at all.

There are infinitely many Carmichael Numbers

- The first three are 561, 1105, 1729

Carmichael Numbers n have the following properties:

- Always odd
- Have at least three prime factors
- Are square free
- If p divides n then $p-1$ divides $n-1$.

Fermat Primality Test
Example: consider $n = 15$.
The values computed for $a^{14} \pmod{15}$ for different values of a are as follows:

a	1	2	3	4	5	6	7	8	9	10	11	12	13	14
$a^{14} \pmod{15}$	1	4	9	1	10	6	4	4	6	10	1	9	4	1

For $a = 1, 4, 11, 14$ the algorithm will output PRIME: these values are called Fermat liars.
For other values of a the algorithm will output COMPOSITE: these values are called Fermat witnesses.

Solovay-Strassen Primality Test
Euler's Criterion: if n is an odd prime and $a \in \mathbb{Z}_n^*$ then:

$$\left(\frac{a}{n}\right) \equiv a^{(n-1)/2} \pmod{n}$$

- $\left(\frac{a}{n}\right)$ is the Jacobi symbol.

- If n is composite then for a random $a \in \mathbb{Z}_n^*$:

$$Pr[\left(\frac{a}{n}\right) = a^{(n-1)/2}] \leq 1/2$$

Algorithm proposed by Solovay and Strassen (1973):

- A randomized algorithm.
- Never incorrectly classifies primes and correctly classifies composites with probability at least 1/2.

Solovay-Strassen Primality Test

```
Pick random a, 1 < a < n−1
if gcd(a,n)>1 then
    return COMPOSITE
end
if (a/n) = a^((n−1)/2) then
    return PRIME
else
    return COMPOSITE
end
```

This test can be repeated k times to reduce the probability of classifying composites as primes.

- If the algorithm outputs COMPOSITE at least once: output COMPOSITE; this will always be correct (a is called an Euler witness).
- If the algorithm outputs PRIME in all the k trials: output PRIME (an Euler pseudoprime); this will be an error with probability $(1/2)^k$.

Solovay-Strassen Primality Test
Example: Consider $n = 15$.
For $a = 3, 5, 6, 9, 10, 12$ the algorithm will output COMPOSITE
For the other values of a which are relatively prime to n:

a	1	2	4	7	8	11	13	14
$\left(\frac{a}{15}\right)$	1	1	1	-1	1	-1	-1	-1
$a^7 \pmod{15}$	1	8	4	13	2	11	7	14

For $a = 1, 14$ the algorithm will output PRIME: these values are called Euler liars.
For other values of a the algorithm will output COMPOSITE: these values are called Euler witnesses.

Miller-Rabin Primality Test
Let 2^k be the largest power of 2 dividing $n-1$.
Thus we have $n-1 = 2^k m$ for some odd number m.
Consider the sequence: $a^{n-1} = a^{2^k m}, a^{2^{k-1}m}, \ldots, a^m$.
We have set this sequence up so that each member of the sequence is a square root of the preceding member.
If n is prime, then by Fermat's Little Theorem, the first member of this sequence $a^{n-1} \equiv 1 \pmod{n}$.
When n is prime, the only square roots of 1 (mod n) are ± 1.
Hence either every element of the sequence is 1, or the first member of the sequence not equal to 1 must be -1 ($\equiv n-1 \pmod{n}$).
The Miller-Rabin test works by picking a random $a \in \mathbb{Z}_n$, then checking that the above sequence has the correct form.

Miller-Rabin Primality Test

```
Pick random a,  1 < a < n−1
b = a^m (mod n)
if b ≠ 1 and b ≠ n−1 then
    i=1
    while i < k and b ≠ n−1
        b = b^2 (mod n)
        if b = 1 then
            return COMPOSITE
        end
        i = i+1
    end
    if b ≠ n−1 then
        return COMPOSITE
    end
end
return PRIME
```

Miller-Rabin Primality Test
For any composite n the probability n passes the Miller-Rabin test is at most 1/4. On average it is significantly less.
The test can be repeated k times to reduce the probability of classifying composites as primes.

- If the algorithm outputs COMPOSITE at least once: output COMPOSITE; this will always be correct (a is called a strong witness).
- If the algorithm outputs PRIME in all the k trials: output PRIME (a strong pseudoprime); this will be an error with probability $(1/4)^k$.

Unlike the Fermat test, there are no composites for which no witness exists.

Miller-Rabin Primality Test
Example: Consider $n = 15$.
$n - 1 = 14 = 2 \times 7$, so $k = 1$, $m = 7$.
For $a = 1, 14$ the algorithm will output PRIME: these values are called strong liars.
For other values of a the algorithm will output COMPOSITE: these values are called strong witnesses

Lucas-Lehmer Primality Test
A Mersenne number is an integer of the form $2^k - 1$, where $k \geq 2$.
If a Mersenne number is a prime, then it is called a Mersenne prime.
Subject of the Great Internet Mersenne Prime Search (GIMPS).
The Mersenne number $n = 2^k - 1$ ($k \geq 3$) is prime if and only if the following two conditions are satisfied:

1. k is prime
2. the sequence of integers defined by $b_0 = 4$, $b_{i+1} = (b_i^2 - 2) \pmod{n}$ $(i \geq 0)$ satisfies $b_{k-2} = 0$.

This is the basis of the Lucas-Lehmer Primality Test.

Lucas-Lehmer Primality Test

```
if k has any factors between 2 and √k
   return COMPOSITE
end
b = 4
for i=1 to k-2 do
   b = (b^2 - 2) mod n
end
if b = 0 then
   return PRIME
else
   return COMPOSITE
```

AKS Primality Test
AKS algorithm discovered by Agrawal, Kayal and Saxena in 2002.
Result of many research efforts to find a deterministic polynomial-time algorithm for testing primality.
Based on the following property: if a and n are relatively prime integers with $n > 1$, n is prime iff:

$$(x - a)^n \equiv x^n - a \pmod{n}$$

where x is a variable.
Always returns correct answer.
Polynomial time algorithm, but still too inefficient to be used in practice.

AKS Primality Test

```
if n has the form a^b (b > 1) then
   return COMPOSITE
end
r = 2
while r < n
   if gcd(n,r) ≠ 1 then return COMPOSITE
   if r is a prime > 2 then
      q=largest factor of r-1
      if q > 4*√r*log n and n^((r-1)/q) ≠ 1 (mod r) then
         break
      end
      r = r+1
   end
end
for a=1 to 2*√r*log n do
   if (x-a)^n ≠ x^n - a (mod gcd(x^r - 1, n)) then return COMPOSITE
end
return PRIME
```

Primality Testing in Practice

The Miller-Rabin test is preferable to the Solovay-Strassen test for the following reasons:

- The Solovay-Strassen test is computationally more expensive.
- The Solovay-Strassen test is harder to implement since it also involves Jacobi symbol computations.
- The error probability for Solovay-Strassen is bounded above by $(1/2)^k$, while the error probability for Miller-Rabin is bounded above by $(1/4)^k$.
- From a correctness point of view, the Miller-Rabin test is never worse than the Solovay-Strassen test.

AKS is a breakthrough result: proves that PRIMES $\in$ P.

- Always gives correct results.
- No practical relevance: prohibitively slow run-times.