Number Theory &
Asymmetric Cryptography
Modular Arithmetic
Notations

\[ \mathbb{Z} = \{-\infty, \cdots -2, -1, 0, 1, 2, \cdots, \infty\} \]

\[ \mathbb{Z}_m = \{0, 1, 2, \cdots, m-2, m-1\} \]

\[ a \equiv b \mod m \Rightarrow a = b + km, \text{ integer } k \]
Addition mod m

Given
\[ a \equiv b \mod m \text{ and } c \equiv d \mod m \]
\[ (a+c) \equiv (b+d) \mod m \]
\[ a-b = km, c-d = lm, \text{ integers } k, l \]
\[ (a+c) = b+d + (k+l)m = (b+d) + jm \]

Examples
\[ (25+15) \equiv (1+3) \equiv 0 \mod 4 \]
\[ (25+15) \equiv (4+1) \equiv 5 \mod 7 \]
Multiplication mod $m$

\[ a \equiv b \mod m, \quad c \equiv d \mod m \]
\[ ac \equiv bd \mod m \]
\[ ac = (b+k\cdot m)(d+l\cdot m) = bd + (bl+kd+k\cdot lm)\cdot m \]

Example

\[ 26 \equiv 2 \mod 4, \quad 11 \equiv 3 \mod 4 \]
\[ 26 \cdot 11 = 286 \equiv 2 \mod 4 \]
\[ 2 \cdot 3 = 6 \equiv 2 \mod 4 \]
What about division?
Is division possible in $\mathbb{Z}_m$?
(A long detour necessary at this point)
Group, Abelian Group, Ring and Field
Group, Ring and Field

Group: Addition is closed, and associative, additive identity and additive inverse exist

Abelian group: Addition is also commutative

Ring: Multiplication is closed, associative, commutative, and distributive; multiplicative identity exists

Field: Every element except “additive identity” has multiplicative inverse

In $\mathbb{Z}_m$ additive identity is 0, multiplicative identity is 1. Its a Ring.

Is $\mathbb{Z}_m$ a Field?
Multiplicative Inverse

Consider m = 5, or \{0,1,2,3,4\}

- \[2 \times 3 \equiv 3 \times 2 \equiv 1 \mod 5\]. So 2 ↔ 3
  - (3 and 2 are multiplicative inverses in \(\mathbb{Z}_5\))
- \[4 \times 4 \equiv 1 \mod 5\], or 4 ↔ 4
- Obviously, 1 ↔ 1

\(\mathbb{Z}_5\) is clearly a field (all elements except additive identity have a multiplicative inverse)

Consider m = 6, or \{0,1,2,3,4,5\}

- 5 ↔ 5 , 1 ↔ 1
- No inverses for 2,3 and 4? (why?)
- \(\mathbb{Z}_6\) is not a Field (it is a Ring)
- (As we shall see soon) the reason for this is that 6 is not a prime number
Basic Theorems of Arithmetic

Any number can be represented as a product of powers of primes. Let $p_i$ be the $i^{th}$ prime

$$ n = \prod_{i=1}^{\infty} p_i^{e_i}, e_i \geq 0 $$

If $n = \prod_{i=1}^{\infty} p_i^{n_i}, m = \prod_{i=1}^{\infty} p_i^{m_i}$

$$ \text{lcm}(m, n) = \prod_{i=1}^{\infty} p_i^{\max(n_i, m_i)} $$

$$ \text{gcd}(m, n) = \prod_{i=1}^{\infty} p_i^{\min(n_i, m_i)} $$

$60 = 2^2 \cdot 3^1 \cdot 5^1, 18 = 2^1 \cdot 3^2$

$$ \text{lcm}(60, 18) = 2^2 \cdot 3^2 \cdot 5^1 = 180 $$

$$ \text{gcd}(60, 18) = 2^1 \cdot 3^1 \cdot 5^0 = 6 $$
GCD Definitions

- \( \text{gcd}(m,n) \) represented as \((m,n)\)
  - If \( a = kb \), where \( k \) is an integer, we say \( b \mid a \) (\( b \) divides \( a \))
  - \( s = (m,n) \) is the largest positive integer satisfying \( s \mid m \) and \( s \mid n \)
  - If \( s = (m,n) = 1 \), \( m \) and \( n \) are relatively prime
  - Euclidean algorithm used for finding \( \text{gcd} \)
GCD Properties

- If $(m,n)=1$, $m|a$, $n|a$ then $mn|a$

- Example $(5,6)=1$ : a number divisible by both 5 and 6 has to be divisible by 30

- $(9,6)$ not equal to 1 : a number divisible by both six and 9 does **not** have to be divisible by 54 (example, 18)

- A useful result:
  - If $(m,n)=1$, $x=a \mod m$, and $x=a \mod n$, then $x=a \mod mn$
  - $(x-a)$ divisible by $m$ and $n$. So it has to be divisible by $mn$
Bezout's Representation

\[ s = (a, b) = ia + jb \] (where \( i \) and \( j \) are integers)

\( s \) (the gcd) is the \textit{smallest positive integer} that can be written \textit{as a combination of} \( a \) and \( b \)

**Practical Implications:**

- If coins are minted in only two denominations \( a \) and \( b \) can we accomplish \textit{any} transaction
- How can you measure 1 foot with two sticks – one 9 feet long and the other 7 feet long? We can because \((7, 9) = 1\).
- \( 1 = 4 \times 7 + (-3) \times 9 \)
Euclidean Algorithm

\((a, b) = (b, c) \text{ if } a = qb + c\)

\((a_0, a_1), a_0 > a_1\)

Let \(a_0 = q_1 a_1 + a_2\)

\((a_0, a_1) = (a_1, a_2)\)

Let \(a_1 = q_2 a_2 + a_3\)

\((a_1, a_2) = (a_2, a_3)\)

\(\vdots\)

and so on until we can trivially find the GCD by inspection
Euclidean Algorithm

(457, 283)
Euclidean Algorithm

\( (457, 283) \)

\[ 457 = 1 \times 283 + 174 \quad \text{or} \quad (457, 283) = (283, 174) \]
Euclidean Algorithm

\[(457, 283)\]

\[457 = 1 \times 283 + 174\]

\[283 = 1 \times 174 + 109\]

\[174 = 1 \times 109 + 65\]

\[109 = 1 \times 65 + 44\]

\[65 = 1 \times 44 + 21\]

\[44 = 2 \times 21 + 2\]

\[21 = 10 \times 2 + 1\]
Euclidean Algorithm

\[(457, 283)\]

\[457 = 1 \times 283 + 174\]
\[283 = 1 \times 174 + 109\]
\[174 = 1 \times 109 + 65\]
\[109 = 1 \times 65 + 44\]
\[65 = 1 \times 44 + 21\]
\[44 = 2 \times 21 + 2\]
\[21 = 10 \times 2 + 1\]
\[2 = 2 \times 1 + 0\]

or \[(457, 283) = (2, 1) = 1\]
Euclidean Algorithm

\((457, 283)\)
\[457 = 1 \times 283 + 174\]
\[283 = 1 \times 174 + 109\]
\[174 = 1 \times 109 + 65\]
\[109 = 1 \times 65 + 44\]
\[65 = 1 \times 44 + 21\]
\[44 = 2 \times 21 + 2\]
\[21 = 10 \times 2 + 1\]
\[1 = 21 - 10 \times 2\] (Bezout Representation)

\[2 = 2 \times 1 + 0\]

\((457, 283) = (283, 174) = (174, 109) = (109, 65) = (65, 44) = (44, 21) = (21, 2) = (2, 1) = 1\]
Backtracking for Bezout

\[(457, 283)\]
457 = 1*283+174
283 = 1*174+109
174 = 1*109+65
109 = 1*65 + 44
65 = 1*44 + 21
44 = 2*21 + 2 \quad 1 = 21-10*(44-2*21)
21 = 10*2 + 1 \quad 1 = 21-10*2
2 = 2*1 + 0 \quad \text{or} \quad (457,283) = (2,1) = 1
Backtracking Euclid

\[(457, 283)\]

\[457 = 1 \times 283 + 174\]
\[1 = 135 \times 457 + (-218) \times 283\]

\[283 = 1 \times 174 + 109\]

\[174 = 1 \times 109 + 65\]

\[109 = 1 \times 65 + 44\]

\[65 = 1 \times 44 + 21\]

\[44 = 2 \times 21 + 2\]
\[1 = 21 - 10 \times (44 - 2 \times 21)\]

\[21 = 10 \times 2 + 1\]
\[1 = 21 - 10 \times 2\]

\[2 = 2 \times 1 + 0\]

or \((457, 283) = (2, 1) = 1\)
Modular Inverse

Does inverse of a exist in $\mathbb{Z}_m$?

$$aa^{-1} \equiv 1 \mod m$$

Let $b = a^{-1}$

$$ab \equiv 1 \mod m \Rightarrow ab = 1 + km \Rightarrow 1 = (-b)a + km$$

$$(a, m) = 1$$

Inverse exists only if $(a, m) = 1$

Else we can not get a Bezout representation linking 1, a and m (like $1 = xa + ym$)

If $(a, m) = 1$ we can use extended Euclidean algorithm to find the inverse of a in $\mathbb{Z}_m$
Multiplicative Inverse of 283 in $\mathbb{Z}_{457}$

$(457, 283)$

$457 = 1 \cdot 283 + 174$  \quad 1 = 135 \cdot 457 + (-218) \cdot 283$

$283 = 1 \cdot 174 + 109$

$174 = 1 \cdot 109 + 65$

$109 = 1 \cdot 65 + 44$

$65 = 1 \cdot 44 + 21$

$44 = 2 \cdot 21 + 2$  \quad 1 = 21 - 10 \cdot (44 - 2 \cdot 21)$

$21 = 10 \cdot 2 + 1$  \quad 1 = 21 - 10 \cdot 2$

$2 = 2 \cdot 1 + 0$  \quad \text{or}  \quad (457, 283) = (2, 1) = 1
Multiplicative Inverse of 283 in $\mathbb{Z}_{457}$

$(457, 283)$

$457 = 1 \cdot 283 + 174$  \quad 1 = 135 \cdot 457 + (-218) \cdot 283$

$283 = 1 \cdot 174 + 109$  \quad (-218 \cdot 283) = 1 + (-135) \cdot 457$

$174 = 1 \cdot 109 + 65$  \quad (-218 \cdot 283) \equiv 1 \pmod{457}$

$109 = 1 \cdot 65 + 44$  \quad -218 \equiv 239 \pmod{457}$

$65 = 1 \cdot 44 + 21$  \quad (239 \cdot 283) \equiv 1 \pmod{457}$

$44 = 2 \cdot 21 + 2$

$21 = 10 \cdot 2 + 1$  \quad 1 = 21 - 10 \cdot 2$

$2 = 2 \cdot 1 + 0$  \quad or $(457, 283) = (2, 1) = 1$
Multiplicative Inverse of 283 in $\mathbb{Z}_{457}$

$$(457, 283)$$

$457 = 1 \cdot 283 + 174 \quad 1 = 135 \cdot 457 + (-218) \cdot 283$

$283 = 1 \cdot 174 + 109 \quad (-218 \cdot 283) = 1 + (-135) \cdot 457$

$174 = 1 \cdot 109 + 65 \quad (-218 \cdot 283) \equiv 1 \mod 457$

$109 = 1 \cdot 65 + 44 \quad -218 \equiv 239 \mod 457$

$65 = 1 \cdot 44 + 21 \quad (239 \cdot 283) \equiv 1 \mod 457$

239 is the inverse of 283 mod 457

Check: $239 \cdot 283 = 67637 = 1 + 148 \cdot 457$
Extended Euclid Algorithm

d=(a,b)=ak+bl
EE Algorithm returns d,k,l
u=[a 1 0];
v=[b 0 1];
while (v(0) != 0) do
  y=floor(u(0)/v(0));
  w=u-y*v;
  u=v; v=w;
endwhile

d=u(0); k=u(1); l=u(2);

\[
\begin{array}{cccccc}
 a&=457, b&=283 \\
 u(0) & u(1) & u(2) & v(0) & v(1) & v(2) \\
 457 & 1 & 0 & 283 & 0 & 1 \\
 283 & 0 & 1 & 174 & 1 & -1 \\
 174 & 1 & -1 & 109 & -1 & 2 \\
 109 & -1 & 2 & 65 & 2 & -3 \\
 65 & 2 & -3 & 44 & -3 & 5 \\
 44 & -3 & 5 & 21 & 5 & -8 \\
 21 & 5 & -8 & 2 & -13 & 21 \\
 2 & -13 & 21 & 1 & 135 & -283 \\
 1 & 135 & -218 & 0 & -283 & 457 \\
\end{array}
\]

1=135*457 + (-218)*283
End of Detour

Now we know why inverses do not exist for 2, 3 and 4 in $\mathbb{Z}_6$

Note that $(5, 6) = 1$

(That's why 5 has an inverse in $\mathbb{Z}_6$)
Prime Modulus

- What if m is prime?
- We have $Z_m = \{0,1,2,...,m-1\}$
- Every number is relatively prime to a prime number
- So every number 1 ... m-1 has an inverse
- $Z_m$ is a FIELD if m is prime
- Normally we call it prime field $Z_p$
RECAP

$Z_m = \{0,1,2,...,m-1\}$

$Z_m$ is a group, and ring

Multiplicative inverse of $a$ exists only if $(a,m)=1$;
GCD computed using Euclidean algorithm
Multiplicative Inverse using Extended Euclidean Algorithm

If $m = p$ (a prime) then $Z_p$ is a field

All elements (except additive identity 0) have a multiplicative inverse.
Why Modular Operations?

• No round off errors
  – For asymmetric cryptography we normally work with very large numbers (several hundred digits each)
  – We will frequently perform computations like $a^b \mod m$ where $a,b$, and $m$ are large numbers
  – All computations will involve repeated modular multiplications (one multiplication followed by one division by the modulus $m$, to get the remainder)
  – If modulus $m$ is a $n$-bit number the largest possible value after a multiplication is guaranteed to be less than $2^n$-bits. After modular reduction the remainder is at most $n$ bits long.
Square and Multiply Algorithm for Exponentiation

Compute $y = a^x \mod n$
Get binary representation of $x$
Let $b(r)b(r-1)\ldots b(0)$ represent bits of $x$

$z=1$;
for $i=r$ downto 0
  $z=z\cdot z \mod n$
  if ($b(i)==1$)
    $z=z\cdot a \mod n$
  endif
endfor
Example

Compute $y = 36^{43} \mod 87$

$43 = 101011_b; r=5; a=36; n=87; z=1$

<table>
<thead>
<tr>
<th>Step</th>
<th>Beginning</th>
<th>End of step</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>$b_5=1, z=1$</td>
<td>$z = z^2 \cdot a = 36 \mod 87$</td>
</tr>
<tr>
<td>2.</td>
<td>$b_4=0, z=36$</td>
<td>$z = z^2 = 78 \mod 87$</td>
</tr>
<tr>
<td>3.</td>
<td>$b_3=1, z=78$</td>
<td>$z = z^2 \cdot a = 45 \mod 87$</td>
</tr>
<tr>
<td>4.</td>
<td>$b_2=0, z=45$</td>
<td>$z = z^2 = 24 \mod 87$</td>
</tr>
<tr>
<td>5.</td>
<td>$b_1=1, z=24$</td>
<td>$z = z^2 \cdot a = 30 \mod 87$</td>
</tr>
<tr>
<td>6.</td>
<td>$b_0=1, z=30$</td>
<td>$z = z^2 \cdot a = 36 \mod 87$</td>
</tr>
</tbody>
</table>

$36^{43} = 36 \mod 87$
Why Prime Modulus?

- It is a Field!
- Rich mathematical operations
- We can pretty much solve any equation as we do in regular math (without concern of overflow)
  - Even square roots
Public Key Cryptography

• Symmetric cryptography requires every pair of interacting entities to share a secret

• Asymmetric cryptography is useful in establishing pair-wise secrets

• Every entity begins by generating a key pair
  – Let Alice's key pair be \((R_A, U_A)\)
    • The private key \(R_A\) is a secret known only to Alice
    • The public key \(U_A\) is made known to everyone
Public key Algorithms

- Two algorithms (E() and D()) were defined for symmetric ciphers
- A “public key algorithm” (for example, RSA) refers to a “system” with up to 6 different sub-algorithms
  - Key pair generation algorithm
  - Encryption algorithm
  - Decryption algorithm
  - Signing algorithm
  - Verification algorithm
  - Key-exchange algorithm
- Not all 6 by necessary for every public-key system
  - Key pair generation is necessary for all.
  - Some do only encryption and decryption,
  - some do only signing/verification,
  - some do only key-exchange
Key Generation Algorithm

- Alice chooses a random private key $R_A$
- And computes the corresponding public $U_A$
- It should be impractical for anyone to compute the private key given the public key
  - Even while the relationship between $(R_A, U_A)$ is one-to-one
  - Example:
    - Private key is two randomly chosen large primes $p$ and $q$
    - The public key is $n=pq$
Encryption Algorithm

• Any one with knowledge of Alice's public key can send a secret message to Alice

• To send a secret $P$, the sender (say, Bob)
  – Uses an encryption algorithm $C = Enc(P, U_A)$
  – Cipher-text $C$ sent to Alice
Decryption Algorithm

• Only the entity with the corresponding private key can decrypt the cipher-text

• \( C = Enc(P, U_A) \) can be decrypted using the \( Dec() \) algorithm as
  
  - \( P = Dec(C, R_A) \)
  
  - As only Alice is privy to her private key \( R_A \), only Alice can decrypt \( C \)
  
  - Can Alice confirm the identity of the sender?
Signing/Verification Algorithms

- To sign a message (typically the hash of a message) Alice uses the signing algorithm $\text{Sig}()$
- Let $M$ be the message hash to be signed
- The signature $S$ is computed as
  - $S = \text{Sig}(M, R_\alpha)$
- Anyone with access to Alice's public key can verify the signature using algorithm $\text{Ver}()$
- Algorithm $\text{Ver}(M, S, U_\alpha)$ returns TRUE only if $S$ was computed using the corresponding private key, and the same message.
- If TRUE, the verifier concludes that the message was signed by Alice.
Inputs and Outputs

- In symmetric cryptography the inputs and outputs (to $E()/D()$) were bit-strings of fixed sizes, with no special interpretation.
- In asymmetric cryptography the inputs and outputs to the five algorithms are seen as numbers belonging to a finite field/ring/group.
- The sizes of inputs/outputs are also more substantial (typically few thousand bits).
Field, Ring, and Group

- Finite sets (of numbers/elements)

- Group:
  - Addition is closed, associative
  - Additive inverse exists; additive identity exists

- Abelian Group are subsets of Groups, where
  - Addition is also commutative

- Rings are subsets of Abelian Groups
  - Multiplication is closed, associative, distributive, and commutative
  - A multiplicative identity exists

- Fields are subsets of Rings
  - Every element except the additive identity has a *multiplicative inverse*
Rings and Fields

• We limit ourselves to one type of finite sets
  – Set of integers modulo m
  – $\mathbb{Z}_m = (0,1,2,...,m-1)$
  – With \textit{modular} addition and multiplication
  – $\mathbb{Z}_m$ is a
    • Field if $m$ is prime;
    • Ring if $m$ is not prime
Examples of Modular Operations in $Z_7$

$Z_7 = \{0, 1, 2, 3, 4, 5, 6\}$

$5 + 6 \equiv 4 \pmod{7}$

$5.6 \equiv 2 \pmod{7}$

$4 + 0 \equiv 4 \pmod{7}$ (0 is additive identity)

$3.1 \equiv 3 \pmod{7}$ (1 is multiplicative identity)

$5.3 \equiv 1 \pmod{7}$ (3 and 5 are multiplicative inverses)

$17^3 \equiv 3^3 \equiv 6 \pmod{7}$
Prime Numbers

• Prime Number Theorem
  – Number of prime numbers less than \( n \) is \( \text{approx. } n / \log(n) \)
  – \( \Psi(n) \approx n / \log(n) \)
  – Prime numbers are \textit{dense}

• Any number can be represented as a product of powers of primes
  – \( n = \prod_{i=1}^{\infty} p_i^{a_i} \)
  – For example
    \[
    10 = 2^1 3^0 5^1
    \]
    \[
    12 = 2^2 3^1
    \]
    \[
    148 = 2^2 \ldots 37^1
    \]
Euler Totient

- Two numbers are **relatively prime** if they have no common factors
  - their GCD is 1.
  - example (9,14)=1,(7,22)=1, etc
- Positive integers $<n$ (or elements of $\mathbb{Z}_n$) that are **relatively prime** to $n$ are **totatives** of $n$
- The **Euler Totient Function** $\Phi(n)$ is the **count** of the totatives of $n$

$$\text{If } n = \prod_{i=1}^{\infty} p_i^{a_i} \text{ then } \Phi(n) = \prod_{i=1}^{\infty} p_i^{a_i - 1} (p_i - 1)$$
Euler Totient

\[ n = \prod_{i=1}^{\infty} p_i^{a_i} \quad \Phi(n) = \prod_{i=1}^{\infty} p_i^{a_i-1}(p_i-1) \]

\[ n = 12 = 2^2 3 \quad \Phi(12) = 2^1(2-1)3^0(3-1) = 4 \]

\[ n = 15 = 3^1 5^1 \quad \Phi(15) = (3-1)(5-1) \]

\[ n = 13 = 13 \quad \Phi(13) = (13-1) \]

• We are particularly interested in two cases
  • Case 1: \( n \) is a prime \( p \)
    • the Euler totient is \( p-1 \)
    • all elements in \( Z_p \) except 0 are totatives
  • Case 2:
    • \( n=pq \) where \( p \) and \( q \) are primes:
      • the Euler totient is \( (p-1)(q-1) \)
      • Only \( (p-1)(q-1) \) of \( n=pq \) elements of \( Z_p \) are totatives
Example, \( m=35 = 5 \times 7 \)

- \( \mathbb{Z}_m = \{0, 1, 2, 3, \ldots, 34\} \)
- 7 multiples of 5, and 5 multiples of 7, are obviously not totatives
- 7 multiples of 5 are 0, 5, 10, 15, 20, 25, 30
- 5 multiples of 7 are 0, 7, 14, 21, 28
- 0 is in both
- \( 7+5-1 = 11 \) numbers are not totatives
- \( 35-11 = 24 = (5-1)(7-1) \) numbers are totatives
- \( \{1, 2, 3, 4, 6, 8, 9, 11, 12, 13, 16, 17, 18, 19, 22, 23, 24, 26, 27, 29, 31, 32, 33, 34\} \) are the 24 totatives of 35.
For General $m=pq$

- More generally
- $p$ multiples of $q$ are not totatives
- $q$ multiples of $p$ are not totatives
- 0 is in both
- $p+q-1$ are not totatives
- $pq-(p+q-1) = (p-1)(q-1)$ numbers are totatives
Multiplicative Inverses

- All totatives of \( \mathbb{Z}_m \) have multiplicative inverses modulo \( m \)
- Totatives of 35
  \{01,02,03,04,06,08,09,11,12,13,16,17,18,19,22,23,24,26,27,29,31,32,33,34\}
- (Corresponding) Multiplicative inverses
  \{01,18,12,09,06,22,04,16,03,27,11,33,02,24,08,32,19,31,13,29,26,23,17,34\}
- \((1,1),(2,18),(3,12)\ldots\) are inverses mod 35
- If \( m \) is prime are numbers in \( \mathbb{Z}_m \) (except 0) have multiplicative inverses
- If \( m=pq \) then only the \((p-1)(q-1)\) totatives have multiplicative inverses

\[
\begin{align*}
4.9 & \equiv 36 \equiv 35.1 + 1 \equiv 1 \mod 35 \\
11.16 & \equiv 176 \equiv 35.5 + 1 \equiv 1 \mod 35 \\
a \cdot a^{-1} & \equiv 35k + 1 \equiv 1 \mod 35 \text{ where } k \text{ is some integer}
\end{align*}
\]
Set of Totatives

Consider the set of $\Phi(m)$ totatives $\{a_1, a_2, \cdots, a_{\Phi(m)}\}$ of $m$.

What happens when you multiply a totative with another? The result has to be a totative. Why?

\[ a_i \cdot a_j = x \] can NOT have any factor common with $m$

$x$ too has to be a totative.

Example 9.12 $\equiv 96 \equiv 3 \pmod{35}$, 19.32 $\equiv 13 \pmod{35}$

Any power of a totative also has to be a totative.

For e.g. in mod 35, $32^2 \equiv 9$, $32^3 \equiv 8$, $32^4 \equiv 11$, $32^5 \equiv 2$
Permutation of Totative Set

Set of $\Phi(m)$ totatives $\{a_1, a_2, \cdots a_{\Phi(m)}\}$ of m
Choose any $a = a_i$ from the set
Multiply every number in the set by $a$ to get $\{a_1 a, a_2 a, \cdots a_{\Phi(m)} a\}$
Now $\{a_1 a, a_2 a, \cdots a_{\Phi(m)} a\}$ is a PERMUTATION of $\{a_1, a_2, \cdots a_{\Phi(m)}\}$.
Why?
Pick two indexes $i, j \in \{1 \cdots \Phi(m)\}$
\[ a_i a \equiv a_j a \Rightarrow (a_i - a_j) a \equiv 0 \mod m \]
Or m divides $(a_i - a_j) a$.
As a has no common factors with m we need m to divide $a_i - a_j$
Which can only happen if $a_i - a_j = 0$ or $i = j$
**Permutation of Totative Set**

Set of $\Phi(m)$ totatives $\{a_1, a_2, \cdots a_{\Phi(m)}\}$ of $m$

Choose any $a = a_i$ from the set

Multiply every number in the set by $a$ to get $\{a_1a, a_2a, \cdots a_{\Phi(m)}a\}$

Now $\{a_1a, a_2a, \cdots a_{\Phi(m)}a\}$ is a PERMUTATION of $\{a_1, a_2, \cdots a_{\Phi(m)}\}$.

Why?

Pick two indexes $i, j \in \{1 \cdots \Phi(m)\}$

$$a_ia \equiv a_ja \Rightarrow (a_i - a_j)a \equiv 0 \mod m$$

Or $m$ divides $(a_i - a_j)a$.

As $a$ has no common factors with $m$ we need $m$ to divide $a_i - a_j$

Which can only happen if $a_i - a_j = 0$ or $i = j$
Permutation of Totative Set

Set of $\Phi(m)$ totatives $\{a_1, a_2, \cdots a_{\Phi(m)}\}$ of $m$

Choose any $a = a_i$ from the set

Multiply every number in the set by $a$ to get $\{a_1a, a_2a, \cdots a_{\Phi(m)}a\}$

Now $\{a_1a, a_2a, \cdots a_{\Phi(m)}a\}$ is a PERMUTATION of $\{a_1, a_2, \cdots a_{\Phi(m)}\}$.

Why?

Pick two indexes $i, j \in \{1 \cdots \Phi(m)\}$

$$a_ia \equiv a_ja \Rightarrow (a_i - a_j)a \equiv 0 \mod m$$

Or $m$ divides $(a_i - a_j)a$.

As $a$ has no common factors with $m$ we need $m$ to divide $a_i - a_j$

Which can only happen if $a_i - a_j = 0$ or $i = j$
Euler Fermat's Theorem

\[ \Phi(m) \text{ Totatives } \{a_1, a_2, \cdots a_{\Phi(m)}\} \text{ of } m \]

\[ \{a_1 a, a_2 a, \cdots a_{\Phi(m)} a\} \text{ is a PERMUTATION of } \{a_1, a_2, \cdots a_{\Phi(m)}\} \]

Multiply all elements together in both sides. The result in both sides should be the same if one side is a permutation of the other

\[ \{a_1 a \times a_2 a \times \cdots \times a_{\Phi(m)} a\} \equiv \{a_1 a_2 \times \cdots \times a_{\Phi(m)}\} \mod m \]

\[ a^{\Phi(m)} \{a_1 a_2 \times \cdots \times a_{\Phi(m)}\} \equiv \{a_1 a_2 \times \cdots \times a_{\Phi(m)}\} \mod m \]

\[ a^{\Phi(m)} \equiv 1 \mod m . \text{ if } a \text{ is relatively prime to } m \]

Fermat's Theorem (special case of Euler-Fermat Theorem)

\[ a^{p-1} \equiv 1 \mod p \text{ if } p \text{ is prime} \]
Euler Fermat's Theorem

Fermat's Theorem

\[ a^{p-1} \equiv 1 \mod p \quad \forall a \]
\[ a^{k(p-1)} \equiv 1 \mod p \quad \forall a \]
\[ a^{k(p-1)+1} \equiv a \mod p \quad \forall a \]

Euler Fermat's Theorem

\[ a^{\varphi(m)} \equiv 1 \mod m. \text{ if } (a,m)=1 \]
\[ a^{k \varphi(m)} \equiv 1 \mod m. \text{ if } (a,m)=1 \]
\[ a^{k \varphi(m)+1} \equiv a \mod m. \text{ if } (a,m)=1 \]

However, if \( m = pq \) then we can show that
\[ a^{k \varphi(m)+1} \equiv a \mod m, \text{ EVEN IF } (a,m) \neq 1 \]
Proof

\[ a^{k \Phi(m)+1} \equiv a \mod m \quad \forall a \text{ if } \Phi(m) = (p-1)(q-1). \]

\[ a^{k \Phi(m)+1} = a^{k(p-1)(q-1)+1} \equiv a \mod p \]

\[ a^{k \Phi(m)+1} = a^{k(p-1)(q-1)+1} \equiv a \mod q \]

Thus, \[ a^{k \Phi(m)+1} \equiv a \mod m = pq \]
Summary

\[ a^{k\Phi(m)+1} \equiv a \mod m \text{forall } a : \text{Useful for RSA} \]
\[ a^{p-1} \equiv 1 \mod p \text{ Fermat's Little Theorem} \]
Simplifying Modular Expressions

Given \( x^y \mod m \), simplifying involves removing all multiples of \( m \) from \( x \) removing all multiples of \( \Phi(m) \) from the exponent \( y \)

\[
x^y \mod m = (x \mod m)^{y \mod \Phi(m)} \mod m
\]

Examples

\[
241^{456} \mod 23 \\
\Phi(23)=22; \ 241 \mod 23 = 11; \ 456 \mod 22 = 16 \\
241^{456} \mod 23 = 11^{16} \mod 23
\]

\[
241^{456} \mod 119 \quad (\Phi(119=7.17)=6.16=96) \\
241 \mod 119 = 3; \ 456 \mod 96 = 72 \\
241^{456} \mod 119 = 3^{72} \mod 119
\]
Difficult Inverse Problems

Factorization Problem
Given two primes $p, q$ trivial to compute $n = pq$
Difficult to find $p$ or $q$ given $n$

Discrete logarithm Problem
prime $p, g$ generator of $Z_p$
Given $p, g, \alpha \equiv g^a \mod p$
Infeasible to find $a$
Choosing a Prime

We know factorization is a difficult problem
Given a number $p$
Can we say if it is a prime w/o factorizing? Yes
How? Probabilistic Primality Testing
We can confirm if a number is a prime with
a vanishingly small probability of failure

Choosing a prime is NOT a hard problem
Probabilistic Primality Check (PPC)

Given a number $p'$
If $p'$ is a prime we know
$$a^{p'-1} \mod p' \equiv 1 \forall a$$
But, if $a^{p'-1} \mod p' \equiv 1$ for some $a$
we can not say it will be true for all $a$
Solution: Choose $n$ random $a$'s for the test
if the test passes every time
probability that $p'$ is NOT a prime is $<1/2^n$
Why?
If the test fails for even for one $a \in \mathbb{Z}_p$, it should fail for
atleast half the possible elements in $\mathbb{Z}_p$. 
Assume test passes for $a = a_1 \cdots a_n$ and fails for $a = x$

$$a_1^{p'-1} \equiv a_2^{p'-1} \equiv \cdots \equiv a_n^{p'-1} \equiv 1 \mod p'$$
$$x^{p'-1} \not\equiv 1 \mod p'$$

In this case the test should also fail for

$$x a_1, x a_2, \cdots, x a_n$$

If the test fails even once it should fail at least as many times as it passes!

The probability of false choice with one test is $1/2$ with $n$ tests it reduces to $1/2^n$
PPC Algorithm

IsPrime(p)
   count =0
   while (count < n)
      a=rand()
      if $a^p \mod p \equiv a$
         count+=1
      else
         break
      endif
   endwhile
   if (count=n) return TRUE
Algorithm to Pick n-digit Prime

Generate random n-bit odd number $p$
while (!IsPrime(p))
    $p+=2; //next odd number$
return $p$
Diffie-Helman Key Exchange

prime $p, g \in \mathbb{Z}_p$ generator: both known to all
Alice chooses random $a \in \mathbb{Z}_p$
Bob chooses random $b \in \mathbb{Z}_p$
Alice to Bob $\alpha \equiv g^a \mod p$
Bob to Alice $\beta \equiv g^b \mod p$
Alice computes $K_{AB} \equiv \beta^a \equiv g^{ab}$
Bob computes $K_{AB} \equiv \alpha^b \equiv g^{ab}$
RSA

Key Pair Generation
1. Choose two large primes $p, q$
2. Compute $n = pq, \varphi(n) = (p - 1)(q - 1)$
3. Choose a small $e$ such that $(e, \varphi(n)) = 1$
4. Find $d$ where $de \equiv 1 \mod \varphi(n)$
5. Destroy $p, q, \varphi(n)$
   public (modulus and exponent) $n, e$
   private (exponent) $d$
RSA Encryption/Decryption

Alice's public keys $n$, $e$, private exponent $d$
To send a secret $P$ to Alice, Bob computes
$$C \equiv P^e \mod n$$
Alice decrypts $C$ as
$$P \equiv C^d \mod n$$
Why does this work?
$$C^d \equiv (P^e)^d \equiv P^{ed} \equiv P^{k\varphi(n) + 1} \equiv P \mod n$$
As $d$ and $e$ are multiplicative inverses mod $\varphi(n)$
RSA Signing / Verification

To sign a hash $M$, Alice computes

$$S \equiv M^d \mod n$$

Any one can verify

$$M \equiv S^e \mod n$$
RSA Example

Key Pair Generation
\[ p = 1009, \quad q = 503 \]
\[ n = pq = 507527, \quad \varphi(n) = (p-1)(q-1) = 506016 \]
\[ e = 5, \quad d = 404813 \quad (5 \times 404813 \mod 506016 = 1) \]
public (modulus and exponent) \( n = 507527, e = 5 \)
private (exponent) \( d = 404813 \)

Encryption
\[ P = 423621 \]
\[ C = P^e \mod n = 423621^5 \mod 5070527 = 32110 \]
\[ P = C^d \mod n = 32110^{404813} \mod 5070527 = 423621 \]
El Gamal Key Generation

\[ p, g, \text{ known to all} \]

Alice chooses random private key \( a \)

Public key is \( \alpha = g^a \mod p \)

Example \( p = 79, g = 7 \)

Alice chooses \( a = 43, \alpha \equiv g^a \equiv 7^{43} \equiv 48 \mod 79 \)

Alice's private key 43,

Alice's public key 48
El Gamal Encryption/Decryption

To send a secret value $P$ to Alice
Bob chooses random $k \in \mathbb{Z}_p$

$$C \equiv P \alpha^k \mod p, \quad \mu \equiv g^k \mod p$$

$C, \mu$ sent to Alice who can compute

$$x \equiv \mu^a \equiv \alpha^k \mod p$$

$$P \equiv Cx^{-1} \mod p$$
El Gamal Encryption/Decryption

Example \( p = 79, \ g = 7, \ a = 43, \ \alpha = 48, \ P = 21, \ k = 5 \)

\[ \mu \equiv g^k \equiv 7^5 \equiv 59 \mod 79 \]

\[ \alpha^k \equiv 48^5 \equiv 54 \mod 79 \]

\[ C \equiv P \alpha^k \equiv 21 \cdot 54 \equiv 28 \mod 79 \]

Alice computes \( \mu^a \equiv 59^{43} \equiv 54 \mod p \)

and \( 54^{-1} \equiv 60 \mod p \)

and \( P \equiv C(\mu^a)^{-1} \equiv 28 \cdot 60 \equiv 21 \mod p \)
El Gamal Signing/Verification

To sign a hash $M$ Alice chooses random $k \in \mathbb{Z}_p$

$$\mu \equiv g^k \mod p$$

$$S \equiv (M - a\mu)k^{-1} \mod p - 1$$

Anyone can verify relationship between $(M, S, \mu, \alpha)$

$$\alpha^\mu \mu^S \equiv g^{a\mu} g^{k(M - a\mu)k^{-1}} \equiv g^M \mod p$$

Example $p = 79, a = 43, \alpha = 48, M = 12, k = 5$

$$\mu \equiv g^k \equiv 7^5 \equiv 59 \mod 79$$

$$k^{-1} \equiv 47 \mod p - 1$$

$$S \equiv (M - a\mu)k^{-1} \equiv (12 - 43.59).47 \equiv 41 \mod 78.$$
Complexity

- Computing exponent: Order of $r$ ($r$ to $2r$) modular multiplications (MM), where $r$ is the number of bits in the exponent
- Computing multiplicative inverse ($n$-bit modulus) is order of $n$ MMs
Complexity of El Gamal

- Encryption, Decryption, signing, verification, key generation are all order of $p$ ($p$ is the modulus)
- Typically $p$ is a 1000-2000 bit number
- Each operation will involve a few thousand MMs
Complexity of RSA

- Modulus $n$ is a 1000-2000 bit number
- For encryption and signature verification the exponent (public exponent) is small (typically 3-4 bits)
- Encryption and signature verification will involve a small number (less than 10) MMs
- For decryption and signing (operation with private exponent) we will require order of thousands of MMs
- How about key generation?
  - A lot more expensive
Complexity of RSA Key Generation

- If modulus n is a 2000 bit number it is advisable to choose a 1001 bit and a 999 bit prime for p and q.
- Choosing a prime is a lot more expensive
  - To pick a 1000 bit prime we might have to try a few hundred odd numbers before we land on a prime
  - For most numbers the PPC test will fail in the first attempt
  - For the number that is finally chosen we might do the check a few hundred times
  - As each test involves one exponentiation (order of 1000 MMs) and we have to choose two primes, RSA key generation can be 3 orders of magnitude more expensive than EL Gamal key generation
Summary of Complexity: Increasing Order

- RSA encryption/signature verification, say 3-5 MMs
- RSA decryption/signing, EL Gamal encryption/signing/decryption/verification/key generation are all a thousand times higher – few thousand MMs
- RSA key generation is 1000 times higher than El Gamal – few million MMs
- All of the above are very much practical though (that's why we use them). The truly difficult problems like factorization/solving discrete log will be off the charts.