

CS8792

Cryptography
and Network

Security

Basic

Concepts in
Number
Theory

Unit-III

Prime
Numbers

Euler's
Theorem

Primality
Testing

Chinese
Remainder
Theorem

Discrete
Logarithms

CS8792 Cryptography and Network Security

Basic Concepts in Number Theory

Session Objectives

CS8792

Cryptography
and Network

Security

Basic

Concepts in
Number
Theory

Unit-III

Prime
Numbers

Euler's
Theorem

Primality
Testing

Chinese
Remainder
Theorem

Discrete
Logarithms

- To learn about prime numbers
- To check a number is prime or not
- To learn Chinese remainder theorem

Agenda

CS8792

Cryptography
and Network

Security

Basic
Concepts in
Number
Theory

Unit-III

Prime
Numbers

Euler's
Theorem

Primality
Testing

Chinese
Remainder
Theorem

Discrete
Logarithms

1 Prime Numbers

2 Euler's Theorem

3 Primality Testing

4 Chinese Remainder Theorem

5 Discrete Logarithms

Presentation Outline

CS8792

Cryptography
and Network

Security

Basic
Concepts in
Number
Theory

Unit-III

Prime
Numbers

Euler's
Theorem

Primality
Testing

Chinese
Remainder
Theorem

Discrete
Logarithms

1 Prime Numbers

2 Euler's Theorem

3 Primality Testing

4 Chinese Remainder Theorem

5 Discrete Logarithms

Prime Numbers and Factorization

CS8792

Cryptography
and Network
Security
Basic
Concepts in
Number
Theory

Unit-III

Prime
Numbers

Euler's
Theorem

Primality
Testing

Chinese
Remainder
Theorem

Discrete
Logarithms

- prime numbers only have divisors of 1 and self
- to factor a number n is to write it as a product of other numbers: $n=a \times b \times c$
- note that factoring a number is **relatively hard compared to multiplying the factors together** to generate the number
- the prime factorisation of a number n is when its written as a product of primes eg. $91=7 \times 13$; $3600 = 2^4 \times 3^2 \times 5^2$

$$a = \prod_{p \in P} p^{a_p}$$

- **Relatively Prime Numbers:** two numbers **a**, **b** are relatively prime if they have no common divisors **except 1**

Fermat's Theorem

CS8792

Cryptography
and Network
Security

Basic

Concepts in
Number
Theory

Unit-III

Prime
Numbers

Euler's
Theorem

Primality
Testing

Chinese
Remainder
Theorem

Discrete
Logarithms

Fermat's Theorem:

$$a^{p-1} \bmod p = 1$$

- where **p** is prime and **gcd(a,p)=1** also known as Fermat's Little Theorem
- useful in public key and primality testing

Euler Totient Function $\phi(n)$

CS8792

Cryptography
and Network

Security

Basic

Concepts in
Number
Theory

Unit-III

Prime
Numbers

Euler's
Theorem

Primality
Testing

Chinese
Remainder
Theorem

Discrete
Logarithms

- when doing arithmetic **modulo n**
- **complete set of residues** is: $0..n-1$
- **reduced set of residues** is those numbers (residues) which are **relatively prime to n**
- eg for $n=10$,
complete set of residues is $0,1,2,3,4,5,6,7,8,9$
reduced set of residues is $1,3,7,9$
- **number of elements in reduced set of residues** is called the **Euler Totient Function $\phi(n)$**

Euler Totient Function $\phi(n)$

CS8792

Cryptography
and Network
Security

Basic

Concepts in
Number
Theory

Unit-III

Prime
Numbers

Euler's
Theorem

Primality
Testing

Chinese
Remainder
Theorem

Discrete
Logarithms

- in general need prime factorization, but

Presentation Outline

CS8792

Cryptography
and Network

Security

Basic
Concepts in
Number
Theory

Unit-III

Prime
Numbers

Euler's
Theorem

Primality
Testing

Chinese
Remainder
Theorem

Discrete
Logarithms

1 Prime Numbers

2 Euler's Theorem

3 Primality Testing

4 Chinese Remainder Theorem

5 Discrete Logarithms

Euler's Theorem

CS8792

Cryptography
and Network
Security

Basic
Concepts in
Number
Theory

Unit-III

Prime
Numbers

Euler's
Theorem

Primality
Testing

Chinese
Remainder
Theorem

Discrete
Logarithms

- a generalisation of Fermat's Theorem

$$a^{\phi(n)} \bmod N = 1$$

- where **$\gcd(a, N) = 1$**
- eg. $a=3$; $n=10$; $\phi(10)=4$;
- hence $3^4 = 81 = 1 \bmod 10$
- $a=2$; $n=11$; $\phi(11)=10$;
- hence $2^{10} = 1024 = 1 \bmod 11$

Presentation Outline

CS8792

Cryptography
and Network

Security

Basic
Concepts in
Number
Theory

Unit-III

Prime
Numbers

Euler's
Theorem

Primality
Testing

Chinese
Remainder
Theorem

Discrete
Logarithms

1 Prime Numbers

2 Euler's Theorem

3 Primality Testing

4 Chinese Remainder Theorem

5 Discrete Logarithms

Primality Testing

CS8792

Cryptography
and Network

Security

Basic
Concepts in
Number
Theory

Unit-III

Prime
Numbers

Euler's
Theorem

Primality
Testing

Chinese
Remainder
Theorem

Discrete
Logarithms

- often need to find large prime numbers
- traditionally sieve using trial division
- ie. divide by all numbers (primes) in turn less than the square root of the number
- only works for small numbers
- alternatively can use statistical primality tests based on properties of primes
- for which all primes numbers satisfy property
- but some composite numbers, called pseudo-primes, also satisfy the property

Miller Rabin Algorithm

CS8792

Cryptography
and Network
Security

Basic
Concepts in
Number
Theory

Unit-III

Prime
Numbers

Euler's
Theorem

Primality
Testing

Chinese
Remainder
Theorem

Discrete
Logarithms

- a test based on Fermat's Theorem
- algorithm is:

TEST (n) is:

- 1 Find integers $k, m, k > 0, m$ odd, so that $(n-1)=2^k \cdot m$
- 2 Select a random integer $a, 1 < a < n-1$
- 3 **if** $a^m \bmod n = 1$ **then return** ("maybe prime");
- 4 **for** $j = 0$ **to** $k - 1$ **do**
- 5 **if** $(a^{2^j m} \bmod n = n-1)$ **then return**(" maybe prime ")
- 6 **return** ("composite")

Miller Rabin Algorithm

CS8792

Cryptography
and Network

Security

Basic

Concepts in
Number
Theory

Unit-III

Prime
Numbers

Euler's
Theorem

Primality
Testing

Chinese
Remainder
Theorem

Discrete
Logarithms

Is 561 prime ?

- 1 Find $561 - 1 = 2^k \cdot m$
- 2 Choose a , $1 < a < n - 1$
- 3 Compute $b_0 = a^m \bmod n$
- 4 if $b_0 = +1 \implies n$ is a composite number
 else if $b_0 = -1 \implies n$ may be a prime number
- 5 Compute $b_i = b_{i-1}^2$, check for composite or prime
- 6 Repeat step number 5

Miller Rabin Algorithm

CS8792

Cryptography
and Network
Security

Basic
Concepts in
Number
Theory

Unit-III

Prime
Numbers

Euler's
Theorem

Primality
Testing

Chinese
Remainder
Theorem

Discrete
Logarithms

■ $n=561$

■ $561-1=2^k \cdot m$

$$\frac{560}{2^2}=280 ; \frac{560}{2^3}=140 ; \frac{560}{2^3}=70 ; \frac{560}{2^4}=35 ; \frac{560}{2^5}=17.5$$

■ $560 = 2^4 \cdot 35 ; k=4; m=35$

1 Choose $a=2$

2 $b_0 = 2^{35} \bmod 561 = 263$

3 Is $b_0 = \pm 1 \bmod 561$

4 $b_1 = b_0^2 = 263^2 \bmod 561 = 67$

5 $b_3 = 67^2 \bmod 561 = 1$

561 is a composite number

Solve: Is 53 a prime number?

Probabilistic Considerations

CS8792

Cryptography
and Network
Security

Basic
Concepts in
Number
Theory

Unit-III

Prime
Numbers

Euler's
Theorem

Primality
Testing

Chinese
Remainder
Theorem

Discrete
Logarithms

- if Miller-Rabin returns "composite" the number is definitely not prime
- otherwise is a prime or a pseudo-prime
- chance it detects a pseudo-prime is $< 1/4$
- hence if repeat test with different random a then chance n is prime after t tests is:
- $\Pr(n \text{ prime after } t \text{ tests}) = 1 - 4^{-t}$
eg. for $t=10$ this probability is > 0.99999

Presentation Outline

CS8792

Cryptography
and Network

Security

Basic
Concepts in
Number
Theory

Unit-III

Prime
Numbers

Euler's
Theorem

Primality
Testing

Chinese
Remainder
Theorem

Discrete
Logarithms

1 Prime Numbers

2 Euler's Theorem

3 Primality Testing

4 Chinese Remainder Theorem

5 Discrete Logarithms

Chinese Remainder Theorem

CS8792

Cryptography
and Network

Security

Basic

Concepts in
Number
Theory

Unit-III

Prime
Numbers

Euler's
Theorem

Primality
Testing

Chinese
Remainder
Theorem

Discrete
Logarithms

Chinese Remainder Theorem: If m_1, m_2, \dots, m_k are pairwise relatively prime positive integers, and if a_1, a_2, \dots, a_k are any integers, then the simultaneous congruences

$$x \equiv a_1 \pmod{m_1}, \quad x \equiv a_2 \pmod{m_2}, \quad \dots, \quad x \equiv a_k \pmod{m_k}$$

have a solution, and the solution is unique modulo m , where

$$m = m_1 m_2 \dots m_k.$$

Chinese Remainder Theorem

CS8792

Cryptography
and Network
Security

Basic

Concepts in
Number
Theory

Unit-III

Prime
Numbers

Euler's
Theorem

Primality
Testing

Chinese
Remainder
Theorem

Discrete
Logarithms

To compute $X \pmod{M}$

- first compute all $a_i = A \pmod{m_i}$ separately
- determine constants c_i ,
where $M_i = M/m_i$
- then combine results to get answer using:

$$X \equiv (\sum_{i=1}^k a_i c_i) \pmod{M}$$

$$c_i = M_i \times (M_i^{-1} \pmod{m_i}) \text{ for } 1 \leq i \leq k$$

Chinese Remainder Theorem

CS8792

Cryptography
and Network
Security

Basic
Concepts in
Number
Theory

Unit-III

Prime
Numbers

Euler's
Theorem

Primality
Testing

Chinese
Remainder
Theorem

Discrete
Logarithms

What's x such that:

$$x \equiv 2 \pmod{3}$$

$$x \equiv 3 \pmod{5}$$

$$x \equiv 2 \pmod{7}?$$

$$X \equiv (\sum_{i=1}^k a_i c_i) \pmod{M} ; c_i = M_i \times (M_i^{-1} \pmod{m_i})$$

$$X = a_1.M_1.M_1^{-1} + a_2.M_2.M_2^{-1} + a_3.M_3.M_3^{-1} \pmod{M}$$

$$M_1.M_1^{-1} \equiv 1 \pmod{m_1}$$

Chinese Remainder Theorem

CS8792

Cryptography
and Network

Security

Basic

Concepts in

Number

Theory

Unit-III

Prime
Numbers

Euler's
Theorem

Primality
Testing

Chinese
Remainder
Theorem

Discrete
Logarithms

Using the Chinese Remainder theorem:

$$a_1 = 2 ; a_2 = 3 ; a_3 = 2 ;$$

$$m_1 = 3 ; m_2 = 5 ; m_3 = 7 ;$$

- $M = m_1 \times m_2 \times m_3 = 3 \times 5 \times 7 = 105$

- $M_1 = M/m_1 = 105/3 = 35$

- 2 is an inverse of $M_1 = 35 \pmod{3}$
(since $35 \times 2 \equiv 1 \pmod{3}$)

- $M_1 \cdot M_1^{-1} \equiv 1 \pmod{m_1} \implies 35 \cdot M_1^{-1} \equiv 1 \pmod{3}$

- $\gcd(35,3); \gcd(3,2); \gcd(2,1); \gcd(1,0) = 1$

- $35 = 11 \times 3 + 2 \implies 2 = 35 - 11 \times 3$

- $3 = 1 \times 2 + 1 \implies 1 = 3 - 1 \times 2$

- $1 = 3 - 1 \times 2$
 $= 3 - (35 - 11 \times 3) = -1 \times 35 + 12 \times 3$

- $1 = -1 \times 35 + 12 \times 3; -1 \times 35 \equiv 1 \pmod{3}$
 $\implies 2 \times 35 \equiv 1 \pmod{3}; \mathbf{2}$ is inverse of $\mathbf{35 \pmod{3}}$

Chinese Remainder Theorem

CS8792

Cryptography
and Network
Security

Basic
Concepts in
Number
Theory

Unit-III

Prime
Numbers

Euler's
Theorem

Primality
Testing

Chinese
Remainder
Theorem

Discrete
Logarithms

Using the Chinese Remainder theorem:

- $M_2 = M/m_2 = 105/5 = 21$
 - 1 is an inverse of $M_2 = 21 \pmod{5}$ (since $21 \times 1 \equiv 1 \pmod{5}$)
- $M_3 = M/m_3 = 105/7 = 15$
 - 1 is an inverse of $M_3 = 15 \pmod{7}$ (since $15 \times 1 \equiv 1 \pmod{7}$)
- So , $X \equiv 2 \times 2 \times 35 + 3 \times 1 \times 21 + 2 \times 1 \times 15 = 233 \equiv 23 \pmod{105}$
- So answer: $X \equiv 23 \pmod{105}$

Presentation Outline

CS8792

Cryptography
and Network

Security

Basic
Concepts in
Number
Theory

Unit-III

Prime
Numbers

Euler's
Theorem

Primality
Testing

Chinese
Remainder
Theorem

Discrete
Logarithms

1 Prime Numbers

2 Euler's Theorem

3 Primality Testing

4 Chinese Remainder Theorem

5 Discrete Logarithms

Primitive Root

CS8792

Cryptography
and Network
Security

Basic

Concepts in
Number
Theory

Unit-III

Prime
Numbers

Euler's
Theorem

Primality
Testing

Chinese
Remainder
Theorem

Discrete
Logarithms

- **Primitive root:** if p is prime, then successive powers of a 'generate' the **group mod p**
- these are useful but relatively hard to find

Powers of Mod 19

\mathbb{Z}_{19}

a	a^2	a^3	a^4	a^5	a^6	a^7	a^8	a^9	a^{10}	a^{11}	a^{12}	a^{13}	a^{14}	a^{15}	a^{16}	a^{17}	a^{18}
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	4	8	16	13	7	14	9	18	17	15	11	3	6	12	5	10	1
3	9	8	5	15	7	2	6	18	16	10	11	14	4	12	17	13	1
4	16	7	9	17	11	6	5	1	4	16	7	9	17	11	6	5	1
5	6	11	17	9	7	16	4	1	5	6	11	17	9	7	16	4	1
6	17	7	4	5	11	9	16	1	6	17	7	4	5	11	9	16	1
7	11	1	7	11	1	7	11	1	7	11	1	7	11	1	7	11	1
8	7	18	11	12	1	8	7	18	11	12	1	8	7	18	11	12	1
9	5	7	6	16	11	4	17	1	9	5	7	6	16	11	4	17	1
10	5	12	6	3	11	15	17	18	9	14	7	13	16	8	4	2	1
11	7	1	11	7	1	11	7	1	11	7	1	11	7	1	11	7	1
12	11	18	7	8	1	12	11	18	7	8	1	12	11	18	7	8	1
13	17	12	4	14	11	10	16	18	6	2	7	15	5	8	9	3	1
14	6	8	17	10	7	3	4	18	5	13	11	2	9	12	16	15	1
15	16	12	9	2	11	13	5	18	4	3	7	10	17	8	6	14	1
16	9	11	5	4	7	17	6	1	16	9	11	5	4	7	17	6	1
17	4	11	16	6	7	5	9	1	17	4	11	16	6	7	5	9	1
18	1	18	1	18	1	18	1	18	1	18	1	18	1	18	1	18	1

For the prime number 19 the **primitive roots** are **2, 3, 10, 13, 14 and 15**

Discrete Logarithms

CS8792

Cryptography
and Network

Security

Basic

Concepts in
Number
Theory

Unit-III

Prime
Numbers

Euler's
Theorem

Primality
Testing

Chinese
Remainder
Theorem

Discrete
Logarithms

- The inverse problem to exponentiation is to find the **discrete logarithm of a number modulo p**
- That is to find **i** such that **$b = a^i \pmod{p}$**
- This is written as **$i = \text{dlog}_a b \pmod{p}$**
- If **a** is a primitive root then it always exists, otherwise it may not,
eg.
- The discrete logarithm does not always exist, for instance there is no solution to
 $2^x \equiv 3 \pmod{7}$.
- There is no simple condition to determine if the discrete logarithm exists.
- Whilst exponentiation is relatively easy, **finding discrete logarithms is generally a hard problem**

Discrete Logarithms

CS8792

Cryptography
and Network
Security
Basic
Concepts in
Number
Theory

Unit-III

Prime
Numbers

Euler's
Theorem

Primality
Testing

Chinese
Remainder
Theorem

Discrete
Logarithms

For example, consider Z_{23}

To compute 3^4 in this group, we first compute $3^4=81$, and then we divide 81 by 23, obtaining a remainder of 12. Thus $3^4=12$ in the group Z_{23}^*

Discrete logarithm is just the inverse operation. For example, take the equation $3^k \equiv 12 \pmod{23}$ for k . As shown above $k=4$ is a solution, but it is not the only solution. Since $3^{22} \equiv 1 \pmod{23}$, it also follows that if n is an integer then $3^{4+22n} \equiv 12 \times 1^n \equiv 12 \pmod{23}$. Hence the equation has infinitely many solutions of the form $4+22n$.

Summary

CS8792

Cryptography
and Network
Security

Basic
Concepts in
Number
Theory

Unit-III

Prime
Numbers

Euler's
Theorem

Primality
Testing

Chinese
Remainder
Theorem

Discrete
Logarithms

- concept of groups, rings, fields
- modular arithmetic with integers
- Euclid's algorithm for GCD & Inverse
- finite fields $GF(p)$
- polynomial arithmetic in general and in $GF(2^n)$
- Fermat's and Euler's Theorems
- Primality Testing
- Chinese Remainder Theorem
- Discrete Logarithms