SNS COLLEGE OF ENGINEERING

Kurumbapalayam (Po), Coimbatore – 641 107

An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

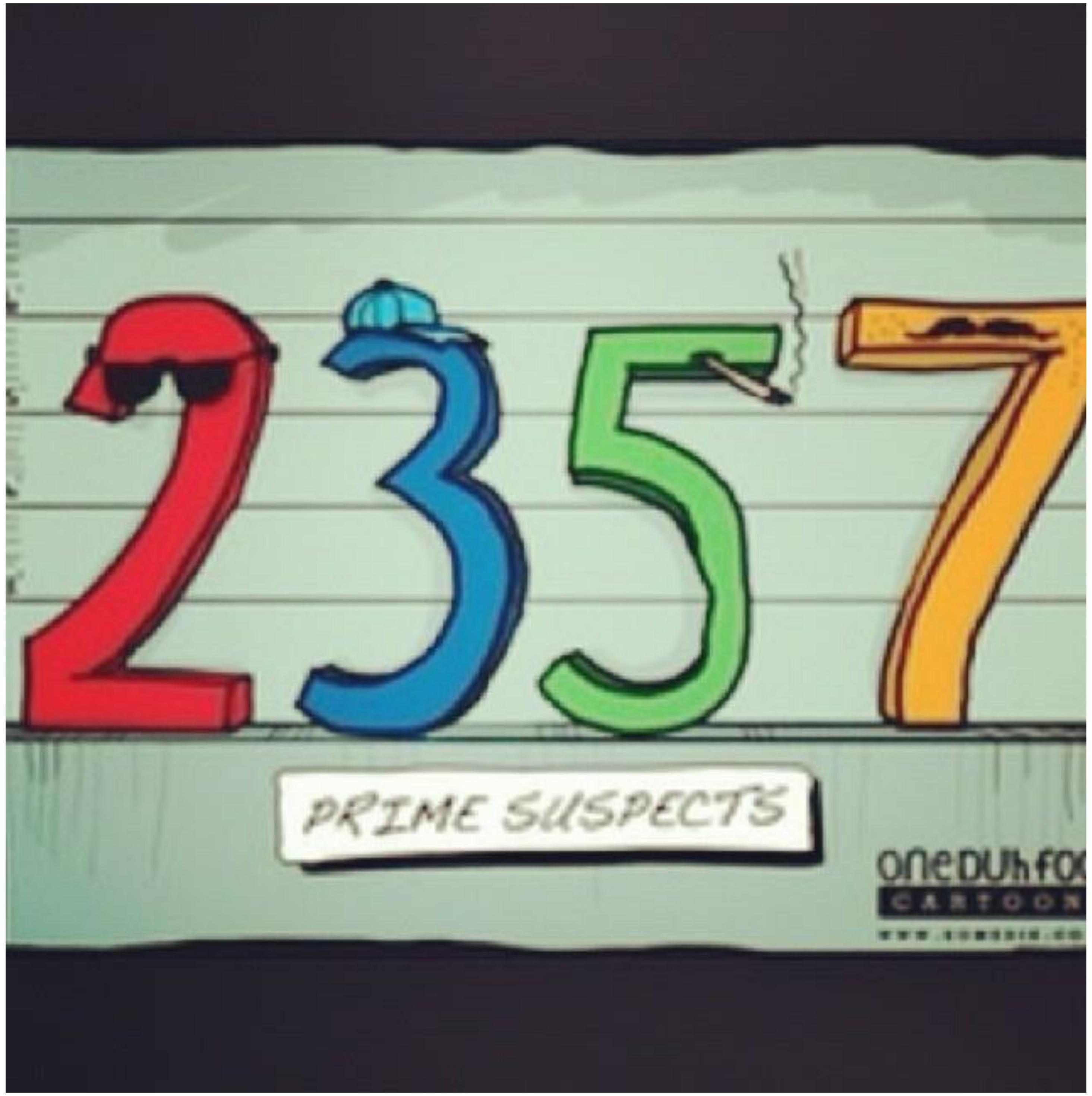
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

COURSE NAME : 19CS503 Cryptography and Network Security

III YEAR V SEMESTER

Unit 3- Public Key Cryptography

Topic : Mathematics of Asymmetric Cryptography: Primes-
Primality Testing


PRIME SUSPECTS

Prime Numbers

- prime numbers only have divisors of 1 and self
  - they cannot be written as a product of other numbers
  - note: 1 is primes generally not of interest me, but
- Example : 2,3,5,7 are prime, 4,6,8,9,10 are not prime numbers are central to number theory
- To factor a number n is to write it as a product of other numbers:

  n=a x b x c
- Prime factorization

Prime Numbers

A prime number has only 2 factors: 1 and itself.

1 is not a prime factor as it has only 1 factor

Prime Numbers

- prime numbers only have divisors of 1 and self
  - they cannot be written as a product of other numbers
  - note: 1 is primes generally not of interest me, but
- Example : 2,3,5,7 are prime, 4,6,8,9,10 are not prime numbers are central to number theory
- To factor a number n is to write it as a product of other numbers:

  n=a x b x c

- Prime factorization

PRIME NUMBERS
BETWEEN 1 AND 1,000

2	79	191	311	439	577	709	857
3	83	193	313	443	587	719	859
5	89	197	317	449	593	727	863
7	97	199	331	457	599	733	877
11	101	211	337	461	601	739	881
13	103	223	347	463	607	743	883
17	107	227	349	467	613	751	887
19	109	229	353	479	617	757	907
23	113	233	359	487	619	761	911
29	127	239	367	491	631	769	919
31	131	241	373	499	641	773	929
37	137	251	379	503	643	787	937
41	139	257	383	509	647	797	941
43	149	263	389	521	653	809	947
47	151	269	397	523	659	811	953
53	157	271	401	541	661	821	967
59	163	277	409	547	673	823	971
61	167	281	419	557	677	827	977
67	173	283	421	563	683	829	983
71	179	293	431	569	691	839	991
73	181	307	433	571	701	853	997

Example

- Factorization
- $91 = 7 * 13$
- $3600 = 2^4 * 3^2 * 5^2$
- $11011 = 7 * 11^2 * 13$

Relatively Prime Numbers & GCD

- Two numbers a, b are relatively prime if have no common divisors apart from 1
- Example:
- 8 & 15 are relatively prime since factors of
  - 8 are 1,2,4,8 and
  - 15 are 1,3,5,15 and 1 is the only common factor

GCD

$300=2^1x3^1x5^2$

$18=2^1x3^2$

$GCD(18,300)=2^1x3^1x5^0=6$

Primality test

- A primality test is an algorithm for determining whether an input number is prime. Among other fields of mathematics, it is used for cryptography.
- Unlike integer factorization, primality tests do not generally give prime factors, only stating whether the input number is prime or not.
- Factorization is thought to be a computationally difficult problem, whereas primality testing is comparatively easy (its running time is polynomial in the size of the input).
- Some primality tests prove that a number is prime, while others like Miller–Rabin prove that a number is composite. Therefore, the latter might more accurately be called compositeness tests instead of primality tests.

Simple methods

- The simplest primality test is trial division: given an input number, n, check whether it is evenly divisible by any prime number between 2 and √n (i.e. that the division leaves no remainder). If so, then n is composite. Otherwise, it is prime.[1]

- For example, consider the number 100, which is evenly divisible by these numbers:

- 2, 4, 5, 10, 20, 25, 50
- Note that the largest factor, 50, is half of 100. This holds true for all n: all divisors are less than or equal to n/2.

- Actually, when we test all possible divisors up to}n/2, we will discover some factors twice. To observe this, rewrite the list of divisors as a list of products, each equal to 100:

$$2 \times 50, 4 \times 25, 5 \times 20, 10 \times 10, 20 \times 5, 25 \times 4, 50 \times 2$$

Simple methods

```
function isPrime(num) {
  if (num <= 3) return num > 1;

  if ((num % 2 === 0) || (num % 3 === 0)) return false;

  let count = 5;

  while (Math.pow(count, 2) <= num) {
    if (num % count === 0 || num % (count + 2) === 0) return false;

    count += 6;
  }

  return true;
}
```

Heuristic tests

- These are tests that seem to work well in practice, but are unproven and therefore are not, technically speaking, algorithms at all. The Fermat test and the Fibonacci test are simple examples, and they are very effective when combined. John Selfridge has conjectured that if p is an odd number, and p ≡ ±2 (mod 5), then p will be prime if both of the following hold:

  2p−1 ≡ 1 (mod p),
  fp+1 ≡ 0 (mod p),
  where fk is the k-th Fibonacci number. The first condition is the Fermat primality test using base 2.

- In general, if p ≡ a (mod x2+4), where a is a quadratic non-residue (mod x2+4) then p should be prime if the following conditions hold:

  2p−1 ≡ 1 (mod p),
  f(1)p+1 ≡ 0 (mod p),
  f(x)k is the k-th Fibonacci polynomial at x.

Assessment 1

1. A non-polynomial function can never agree with euler's theorem.
   a) True
   b) false

2. For homogeneous function with no saddle points we must have the minimum value as ____________
   a) 90
   b) 1
   c) equal to degree
   d) 0

REFERENCES

1. William Stallings, Cryptography and Network Security, 6 th Edition, Pearson Education, March 2013.

THANK YOU