

CSCI 454/554 Computer and Network Security

Topic 5.1 Basic Number Theory -- Foundation of Public Key Cryptography



- GCD and Euclid's Algorithm
- Modulo Arithmetic
- Modular Exponentiation
- Discrete Logarithms

2



WILLIAM & MARY

GCD and **Euclid's** Algorithm



Some Review: Divisors WILLIAM GMARY

- Set of all integers is Z = {...,-2, -1,0,1,2, ...}
- b divides a (or b is a divisor of a) if a = mb for some m
 - denoted b|a
 - any $b \neq 0$ divides 0
- For any a, 1 and a are trivial divisors of a
 - all other divisors of a are called factors of a

4

Primes and Factors WILLIAM & MARY

- a is prime if it has no non-trivial factors
 - examples: 2, 3, 5, 7, 11, 13, 17, 19, 31,...
- Theorem: there are infinitely many primes
- Any integer a>1 can be factored in a unique way as $p_1^{a_1} \bullet p_2^{a_2} \bullet \dots p_t^{a_t}$
 - where all $p_1>p_2>...>p_t$ are prime numbers and where each $a_i>0$

Examples: 91 =

 $91 = 13^{1} \times 7^{1}$

11011 = 13¹ ×11² ×7¹

Ŵ

Common Divisors

WILLIAM GMARY

 A number d that is a divisor of both a and b is a common divisor of a and b

Example: common divisors of 30 and 24 are 1, 2, 3, 6

If d|a and d|b, then d|(a+b) and d|(a-b)

Example: Since 3 | 30 and 3 | 24, 3 | (30+24) and 3 | (30-24)

If d|a and d|b, then d|(ax+by) for any integers x and y

Example: $3 \mid 30 \text{ and } 3 \mid 24 \implies 3 \mid (2*30 + 6*24)$

6

```
Greatest Common Divisor (GCD) WILLIAM

• gcd(a,b) = max\{k \mid k \mid a \text{ and } k \mid b\}

Example: gcd(60,24) = 12, gcd(a,0) = a

• Observations

• gcd(a,b) = gcd(|a|, |b|)

• gcd(a,b) \leq min(|a|, |b|)

• if 0 \leq n, then gcd(an, bn) = n*gcd(a,b)

• For all positive integers d, d, and d...

...if d \mid db

...and gcd(a,d) = 1

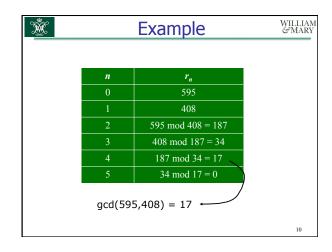
...then d \mid b
```

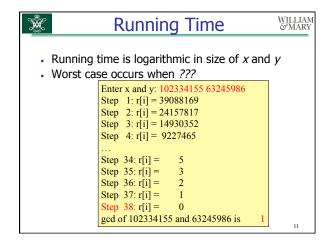
```
• Computing GCD by hand: if a = p_1^{a1} p_2^{a2} \dots p_r^{ar} and b = p_1^{b1} p_2^{b2} \dots p_r^{br}, ...where p_1 < p_2 < \dots < p_r are prime, ...and a_i and b_i are nonnegative, ...then gcd(a, b) = p_1^{\min(a1, b1)} p_2^{\min(a2, b2)} \dots p_r^{\min(ar, br)}

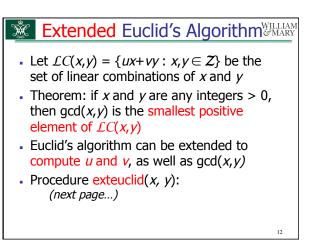
• Slow way to find the GCD

• requires factoring a and b first (which can be slow)
```

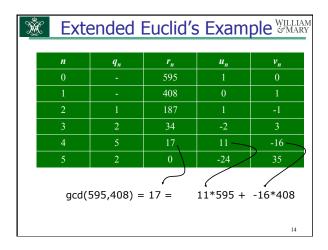
Euclid's Algorithm for GCD WILLIAM (MARY)
 Insight: gcd(x, y) = gcd(y, x mod y)
 Procedure euclid(x, y):
 r[0] = x, r[1] = y, n = 1; while (r[n] != 0) {
 n = n+1;
 r[n] = r[n-2] % r[n-1];
 }
 return r[n-1];

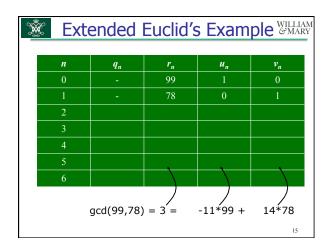


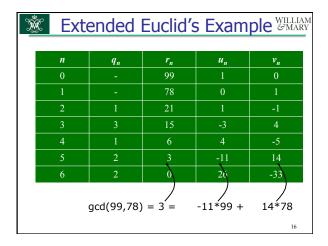


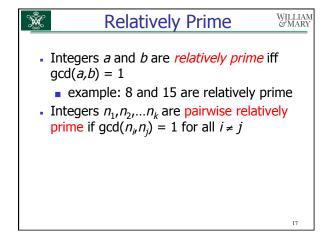


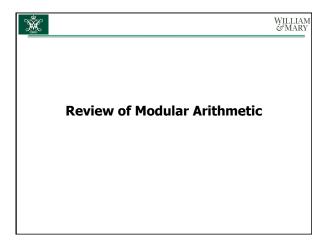
```
| Extended Euclid's Algorithm | Find | Find
```

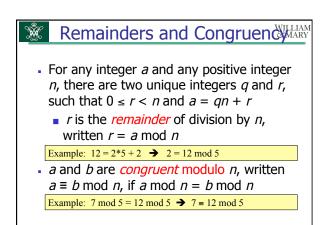


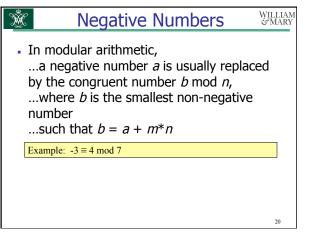


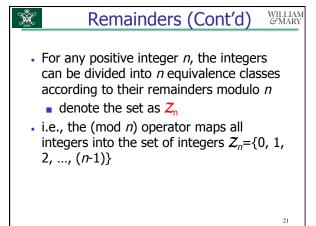


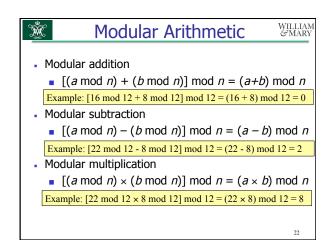


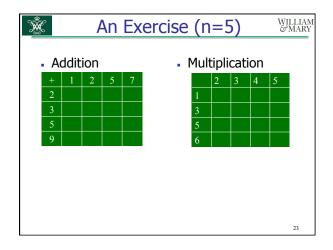


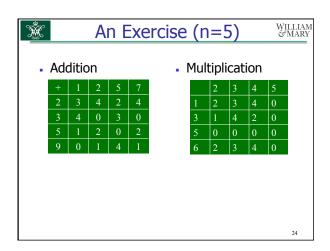


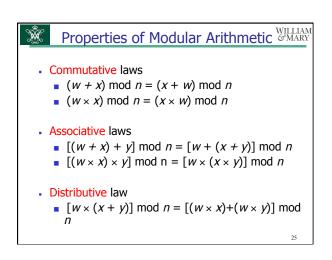


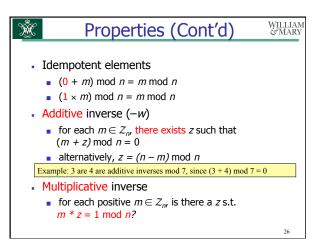


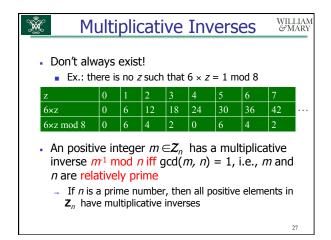


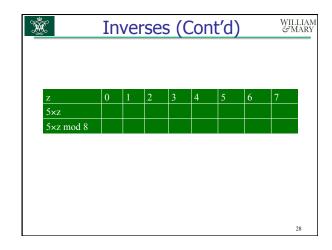


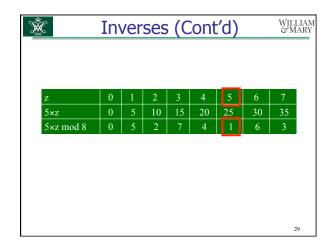


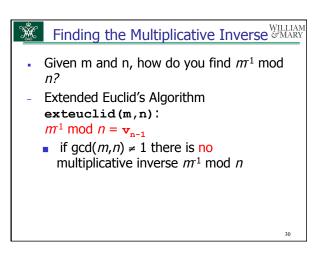


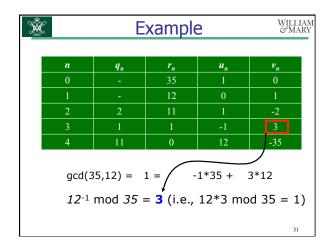


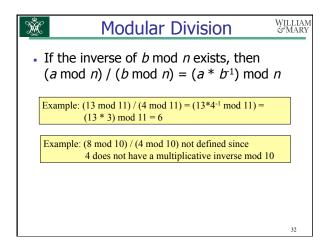


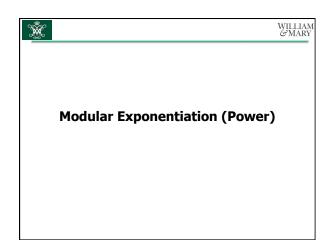


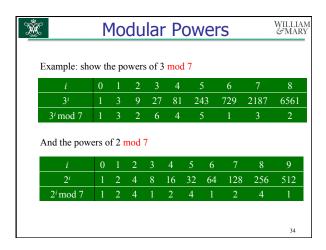


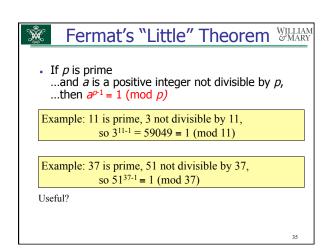


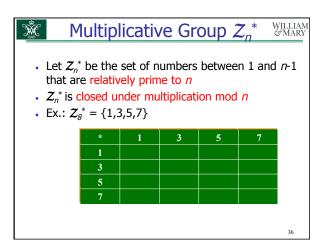


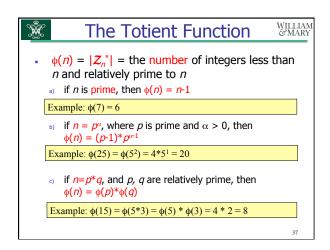


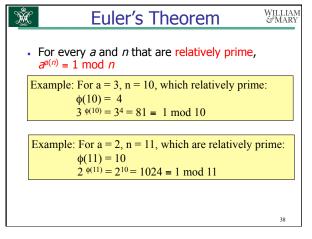


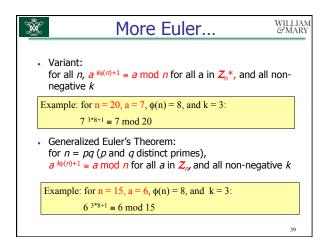


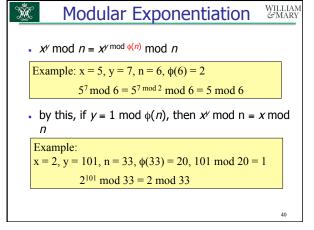


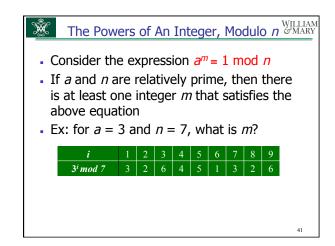


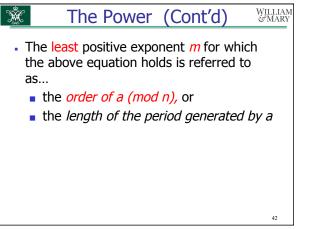


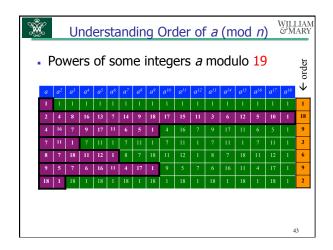


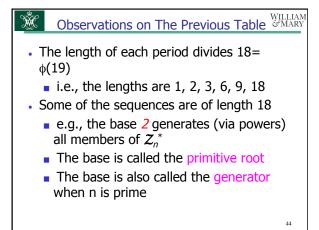


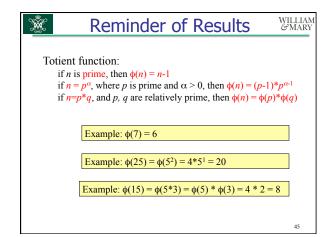


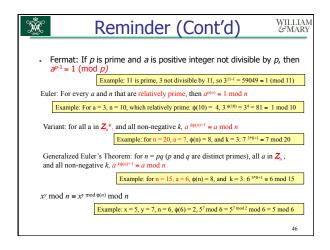


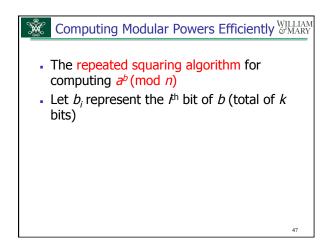


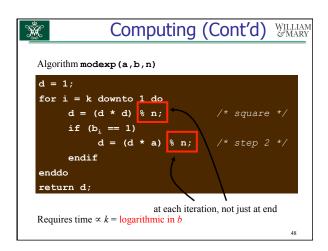


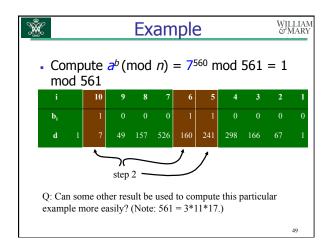


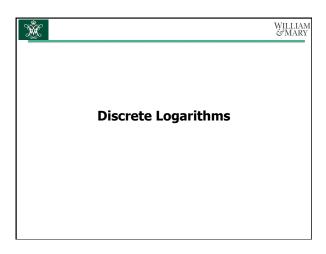




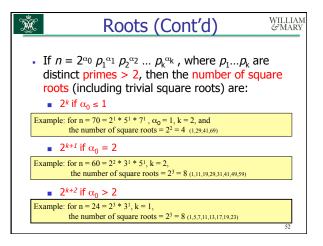








Square Roots WILLIAM EMARY
 x is a non-trivial square root of 1 mod n if it satisfies the equation x² = 1 mod n, but x is neither 1 nor -1 mod n
 Ex: 6 is a square root of 1 mod 35 since 6² = 1 mod 35
 Theorem: if there exists a non-trivial square root of 1 mod n, then n is not a prime
 i.e., prime numbers will not have non-trivial square roots



Primitive Roots
 Reminder: the highest possible order of a (mod n) is φ(n)
 If the order of a (mod n) is φ(n), then a is referred to as a primitive root of n
 for a prime number p, if a is a primitive root of p, then a, a², ..., ap-1 are all distinct numbers mod p
 No simple general formula to compute primitive roots modulo n
 there are methods to locate a primitive root faster than trying out all candidates

