Announcements

- Difference in course logistics from the fall
  - Schedule of class meetings
  - Group Presentations
  - Kryptos competition

- Goals for today
  - Big picture overview of course content
  - Review of EEA
  - Point out slightly different notation from the fall

Big Picture of Course Content

1. Security of DHKE, DSA depend on DLP being hard to solve $g^x \equiv h \mod p$
   Are there non-brute force attacks?
   - Shanks Babystep-Giantstep algorithm
     Requires storing two lists, can become impractical
   - Pollig-Hellman algorithm
     Shows why we want α to have large prime order in $\mathbb{Z}_p^*$
   - Pollard's ρ is a general collision algorithm
     More efficient in storage than Shanks in storage, but runtime may be longer

Big Picture of Course Content

1. Security of DHKE, DSA depend on DLP being hard to solve $g^x \equiv h \mod p$

   Are there non-brute force attacks?

   - Shanks Babystep-Giantstep algorithm
     Requires storing two lists, can become impractical
   - Pollig-Hellman algorithm
     Shows why we want α to have large prime order in $\mathbb{Z}_p^*$
   - Pollard's ρ is a general collision algorithm
     More efficient in storage than Shanks in storage, but runtime may be longer

2. RSA, DHKE, DSA depend on finding large primes
   - How do you do this?
   - Can apply Pollard's ρ to factor integers

Big Picture of Course Content (cont)

4. Elliptic curve cryptography requires dramatically smaller keys than RSA, DHKE, DSA for equivalent level of security

   How does this work?

Big Picture of Course Content (cont)

4. Elliptic curve cryptography requires dramatically smaller keys than RSA, DHKE, DSA for equivalent level of security

   How does this work?

5. Security of these methods fall apart when quantum computers are feasible

   Look at basis for lattice-based encryption schemes which have no known quantum attacks

Big Picture of Course Content (cont)

4. Elliptic curve cryptography requires dramatically smaller keys than RSA, DHKE, DSA for equivalent level of security

   How does this work?

5. Security of these methods fall apart when quantum computers are feasible

   Look at basis for lattice-based encryption schemes which have no known quantum attacks

6. Eight other topics from you!

Proposition 1.7

Let a, b be positive integers where $a \geq b$.
The Euclidean Algorithm computes $\gcd(a, b)$ in at most $2\log_2(b) + 2$ steps.

Example: Find $\gcd(77, 12)$

One consequence of Theorem 1.7

- In RSA, we publish public (n, e) where $\gcd(e, \phi(n)) = 1$
  Then $d = e^{-1} \mod \phi(n)$ is the private key

One consequence of Theorem 1.7

- In RSA, we publish public (n, e) where $\gcd(e, \phi(n)) = 1$
  Then $d = e^{-1} \mod \phi(n)$ is the private key

- If we pick $e = 2^7 + 1 = 129 = 10000001_2$ (so efficient with square & multiply), then

$$2\log_2(129) + 2 \approx 16.02$$

  so will take at most 17 steps to find d no matter how big n is!

Recall basics of modular arithmetic

$a \equiv b \mod m$ iff a and b have the same remainder when divided by m

Equivalently, $a \equiv b \mod m$ iff $m \mid (a - b)$

Notation

- The *ring of integers modulo m* is $\mathbb{Z}/m\mathbb{Z} = \{0, 1, 2, \ldots, m-1\}$

- The the group of units mod m is

$$\begin{aligned}(\mathbb{Z}/m\mathbb{Z})^* &= \{a \in \mathbb{Z}/m\mathbb{Z} \mid a \text{ has a multiplicative inverse } \} \\ &= \{a \in \mathbb{Z}/m\mathbb{Z} \mid \gcd(a, m) = 1\} \\ &= \{a \in \mathbb{Z}/m\mathbb{Z} \mid \text{there exist } u, v \in \mathbb{Z} \text{ such that } au + mv = 1\}\end{aligned}$$

- If p is prime, then $\mathbb{F}_p = \mathbb{Z}/p\mathbb{Z}$ is the finite field of order p
  And $\mathbb{F}_p^* = \{1, 2, \ldots, p-1\}$

1. (a) List the elements of $\mathbb{Z}/8\mathbb{Z}$
   (b) List the elements of $(\mathbb{Z}/8\mathbb{Z})^*$
   (c) Find the order of each element in $(\mathbb{Z}/8\mathbb{Z})^*$
   (d) Find the inverse of each element in $(\mathbb{Z}/8\mathbb{Z})^*$

2. (a) List the elements of $\mathbb{F}_7$
   (b) List the elements of $\mathbb{F}_7^*$
   (c) Find the order of each element in $\mathbb{F}_7^*$
   (d) Find the inverse of each element in $\mathbb{F}_7^*$