Cryptography 1/24

Reagan Shirk

January 24, 2020

(Extended) Euclidean Algorithm

- People say this was the first non-trivial algorithm
- We want to prove that this algorithm is efficient
- Why did someone want to create this algorithm, even if they're only working with small numbers?
- Basic idea: remaindering sequence
  - Start with numbers n and m and you calculate $r_1,\ r_2,\ r_3,\ \cdots,\ r_t = 0$
- Lemma (what we want to prove to prove correctness): If $n = qm + r$, then $\gcd(n, m) = \text{gdc}(m, r)$
  - Reminder: for algorithms we need to prove termination, correctness, and efficiency
- How to we prove the lemma?
  - $a|b$ and $b|a \Rightarrow |a| = |b|$
  - First step: $\gcd(n, m) \mid \text{gdc}(m, r)$
    - $\gcd(n, m) \mid m \to$ trivial because of the definition of divisor
    - $\gcd(n, m) \mid r \to$ because $r = n - qm$ (taken from above lemma), we know that $n|n$ and $m|qm$ so we can prove that $\gcd(n, m) \mid r$
  - Second step: $\text{gdc}(m, r) \mid \gcd(n, m)$
    - Similar process to above
- We want to prove that t is small for…some reason
- Lemma: $t = O(log(n)) \to$ this will show…one of the things we need to prove
- Lemma: $r_{i+2} \leq \frac{r_i}{2},\ r_i = qr_{i+1} + r_{i+2}$
  - proof: case study
    - Case 1:
      - $r_{i+1} < \frac{r_i}{2}$
      - $\Rightarrow r_{i+2} < \frac{r_i}{2}$
    - Case 2:
      - $r_{i+1} \geq \frac{r_i}{2}$
      - $\Rightarrow q = 1$
      - $\Rightarrow r_{i+2} = r_i - r_{i+1} < \frac{r_i}{2}$
    - Case 3:
      - $r_{i+1} = \frac{r_i}{2} \Rightarrow r_{i+2} = 0$

Residue Class Ring

- We have $12\mathbb{Z}$ (all multiples of 12)
- So when we're doing calculations over hours, we do $\frac{\mathbb{Z}}{12\mathbb{Z}}$