# Math 1580: Cryptography *Lecture Notes*

E. Larson

Spring 2022

These are lecture notes for Math 1580: Cryptography taught at Brown University by Eric Larson in the Spring of 2022.

## Contents

# §0 January 26, 2022

## §0.1 Course Logistics

## §0.2 Introduction

## §0.3 Simple Substitution Ciphers

## §0.4 Divisibility

# §1 January 28, 2022

## §1.1 Greatest Common Divisors

## §1.2 Euclidean Algorithm

## §1.3 Linear Combinations

# §2 January 31, 2022

## §2.1 Linear Combinations *continued*

Recall from last time that we proposed that

$$\text{greatest common divisor} \le \text{least linear combination}.$$

**Example 2.1.1**

$\gcd(2024, 748) = 44$ because we have

$$\begin{aligned}
2024 &= 748 \cdot 2 + 528 \\
748 &= 528 \cdot 1 + 220 \\
528 &= 220 \cdot 2 + 88 \\
220 &= 88 \cdot 2 + \boxed{44} \leftarrow \gcd(2024, 748) \\
88 &= 44 \cdot 2 + 0
\end{aligned}$$

We determine which linear combinations or 2024 znd 748 we can create:

$$\begin{aligned}
2024 &= 1 \cdot 2024 + 0 \cdot 748 \\
748 &= 0 \cdot 2024 + 1 \cdot 748 \\
528 &= 1 \cdot 2024 + (-2) \cdot 748 \\
220 &= 748 - 1 \cdot 528 \\
&= 748 - 1 \cdot (1 \cdot 2024 + (-2) \cdot 748) \\
&= -1 \cdot 2024 + 3 \cdot 748 \\
88 &= 528 - 2 \cdot 220 \\
&= \underbrace{[1 \cdot 2024 + (-2) \cdot 748]}_{528} - 2 \cdot \underbrace{[-1 \cdot 2024 + 3 \cdot 748]}_{220} \\
&= 3 \cdot 2024 - 8 \cdot 748 \\
44 &= 220 - 2 \cdot 88 \\
&= [-1 \cdot 2024 + 3 \cdot 748] - 2 \cdot [3 \cdot 2024 - 8 \cdot 748] \\
&= -7 \cdot 2024 + 19 \cdot 748
\end{aligned}$$

Following this example, we have shown that every common divisor of $a$ and $b$ can be written as a linear combination of $a$ and $b$, and since the greatest common divisor has to be less than the least linear combination (as shown last time), the greatest common divisor *is* the least linear combination[1].

We realize that there is a *recurrence* happening here. If we call every set of coefficients $x, y$ and $z, w$ for $a$ and $b$ respectively, such that

$$\begin{aligned}
a &= x \cdot a_0 + y \cdot b_0 \\
b &= z \cdot a_0 + y \cdot b_0
\end{aligned}$$

where $a_0$ and $b_0$ are the original numbers, we can use a sliding window approach[2] again to determine the next set of $x, y, z, w, a, b$.

Recall from last time we had

$$\begin{aligned}
a' &= b \\
b' &= a \mod b
\end{aligned}$$

We can extend this algorithm for our new coefficients:

$$\begin{aligned}
x' &= z \\
y' &= w \\
z' &= w - \left\lfloor \frac{a}{b} \right\rfloor \cdot z \\
w' &= y - \left\lfloor \frac{a}{b} \right\rfloor \cdot w
\end{aligned}$$

[1] Assume for contradiction that the gcd were any less, then that would also be a linear combination. ↯

[2] Updating our iterators on every loop by sliding our window of coefficients down.

where $\lfloor \frac{a}{b} \rfloor$ are the quotients from our Euclidean Algorithm. Note that initially, we have

$$a = 1 \cdot a_0 + 0 \cdot b_0$$
$$b = 0 \cdot a_0 + 1 \cdot b_0$$

so we have initial values of $x = 1, y = 0, z = 0, w = 0$.

so our code for the *extended Euclidean's Algorithm* is now

```
def ext_gcd(a, b):
    x, y, z, w = 1, 0, 0, 1
    while b != 0:
        x, y, z, w = z, w, w - (a // b) * z, y - (a // b) * w
        a, b = b, a % b
    return (x, y)
```

## §2.2 Modular Arithmetic

*Recall:* We used a substitution/shift cipher to encrypt text:

$$\begin{array}{ccc} \texttt{Y} & \texttt{E} & \texttt{S} \\ \downarrow & \downarrow & \downarrow \\ \texttt{D} & \texttt{J} & \texttt{X} \end{array}$$

by incrementing 5 letters for each lecture.

$a = 0,\ b = 1,\ \ldots,\ z = 25$.

We had this notion of

$$\text{ciphertext} = \text{plaintext} + 5$$
$$\texttt{d} = \texttt{y} + 5$$
$$3 = 24 + 5 = 29$$

**Definition 2.2.1.** We say $a \equiv b \mod m$ if $m \mid a - b$.

We say "$a$ is congruent[3] to $b$ modulo $m$".

**Example 2.2.2**

$$24 + 5 \equiv 3 \mod 26$$
$$22 + 2 \equiv 1 \mod 12$$

[3] Congruence is a "behave like" equality.

The first example is from our shift sipher, the second example is equivalent to "two hours after 11:00, it is 1:00".

**Proposition 2.2.3**

If we have

$$a_1 \equiv a_2 \mod m$$
$$b_1 \equiv b_2 \mod m$$

Then we have the following:

$$a_1 + b_1 \equiv a_2 + b_2 \mod m \tag{1}$$
$$a_1 - b_1 \equiv a_2 - b_2 \mod m \tag{2}$$
$$a_1 \cdot b_1 \equiv a_2 \cdot b_2 \mod m \tag{3}$$

*Proof.* For eq. (1), realize that we have

$$(a_1 + b_1) - (a_2 + b_2) = (a_1 - a_2) + (b_1 - b_2)$$

and the two terms on the right are each divisible by $m$ by our premise. We can also write out

$$\begin{aligned} a_1 + b_1 &= (a_2 + \alpha m) + (b_2 + \beta m) \\ &= (a_2 + b_2) + (\alpha + \beta) \cdot m. \end{aligned}$$

Similarly, for eq. (2), we have

$$\begin{aligned} a_1 - b_1 &= a_2 + \alpha m - (b_2 + \beta m) \\ &= a_2 - b_2 + (\alpha - \beta) \cdot m. \end{aligned}$$

and for eq. (3), we have

$$\begin{aligned} a_1 \cdot b_1 &= (a_2 + \alpha m) \cdot (b_2 + \beta m) \\ &= a_2 \cdot b_2 + \alpha m b_2 + \beta m a_2 + \alpha\beta m^2 \\ &= a_2 \cdot b_2 + (\alpha b_2 + \beta a_2 + \alpha\beta m) \cdot m. \end{aligned}$$

which concludes the proofs of the premod rules. □

**Proposition 2.2.4**

There exists $b$ with

$$a \cdot b \equiv 1 \mod m$$

if and only if $\gcd(a, m) = 1$.

*Proof.* We can write linear combination equation

$$a \cdot b + m \cdot k = 1$$

and we have that the following are equivalent (we cascade down the list and can easily prove the iff relations):

i. such a $b$ exists,

ii. there is a solution $b, k$ to this equation,

iii. 1 is a linear combination of $a$ and $m$,

iv. 1 is the *least* linear combination of $a$ and $m$,

v. $1 = \gcd(a, m)$.

so we have that $1 = \gcd(a, m)$ if and only if $a$'s inverse $b$ exists. $\square$