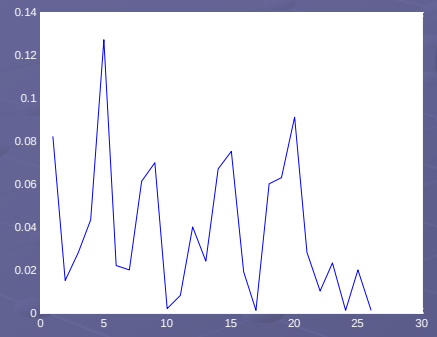
# DTTF/NB479: Dszquphsbqiz Day 6

- Announcements:
  - Programming exam next Thursday on breaking codes from chapter 2
  - Written exam at start of week 4 on concepts from chapter 2
- Questions?

- This week: see schedule page
  - 2 days of chapter 3, then back to Hill cipher

# Vigenere is more secure than affine cipher, but still breakable




- You should be able to answer:
  1. What makes a Vigenere cipher more secure than a shift cipher?
  2. Why does the max of dot($A_0$,$A_i$) occur when i==0?
  3. What are the advantages and disadvantages of using the dot product method (method 2) vs. max is ‘e’ (method 1) to decrypt the key?
  4. How do we find the key length?

# Does combining Vigenere with affine ciphers help?

- *What if we modified the Vigenere cipher so that each individual letter was not simply shifted, but the result of an affine function?*

# Vigenere can be made secure with appropriate precautions

- From http://sharkysoft.com/misc/vigenere/
  - Key must be as long as the plaintext message.
  - Build the key from random characters.
  - Never use the key again.
  - Don't use text decorations (spaces, punctuations, capitalization).
  - Protect the key.

# Vigenere trivia (if time at end)

- Consider *Gadsby* by Ernest Vincent Wright, February 1939:
  - http://www.spinelessbooks.com/gadsby/01.html
- What do you notice about it?

# Back to Basics: 3. GCD

- gcd(a,b)=$\max_j$ (j|a and j|b).
- Def.: a and b are relatively prime iff gcd(a,b)=1
- gcd(14,21) easy…
- What about gcd(1856, 5862)?
- Or gcd(500267500347832384769, 120928345434758932565746 65)?

- Do you really want to factor each one?
- What's our alternative?

# Euclid's Algorithm

```
gcd(a,b) {
    if (a < b) swap (a,b)
    // a > b
    r = a % b
    while (r ~= 0) {
        a = b
        b = r
        r = a % b
    }
    gcd = b // last r ~= 0
}
```

**Calculate `gcd(1856, 5862)`**

=2

# Euclid's Algorithm

```
gcd(a,b) {
    if (a > b) swap (a,b)
    // a > b
    r = a % b
    while (r ~= 0) {
        a = b
        b = r
        r = a % b
    }
    gcd = b // last r ~= 0
}
```

Assume $a > b$

Let $q_i$ and $r_i$ be the series of quotients and remainders, respectively, found along the way.

$$a = q_1 b + r_1$$
$$b = q_2 r_1 + r_2$$
$$r_1 = q_3 r_2 + r_3$$
$$\ldots$$
$$r_{i-2} = q_i r_{i-1} + r_i$$
$$\ldots$$
$$r_{k-2} = q_k r_{k-1} + r_k$$
$$r_{k-1} = q_{k+1} r_k$$

$r_k$ is gcd(a,b)

You'll prove this computes the gcd in Homework 3 (by induction)…

# Fundamental result:
## *If d = gcd(a,b) then ax + by = d*

- For *some* integers x and y.
- ***These ints are just a by-product of the Euclidean algorithm!***
- Allows us to find $a^{-1}$ *(mod n)* very quickly…
  - Choose b = n and d = 1.
  - If gcd(a,n) =1, then ax + ny = 1
  - ax = 1 (mod n) because it differs from 1 by a multiple of n
  - Therefore, $x = a^{-1}$ (mod n).
- Why does the result hold?
- How do we find x and y?

# *Why does this work?*

*Given a,b ints, not both 0, and gcd(a,b) = d.*

*Prove ax + by = d*

Recall gcd(a,b,)=d = $r_k$ is the last non-zero remainder found via Euclid.

We'll show the property true for all remainders $r_j$ (by strong induction)

**Assume a > b**

**Let $q_i$ and $r_i$ be the series of quotients and remainders, respectively, found along the way.**

$$a = q_1 b + r_1$$

$$b = q_2 r_1 + r_2$$

$$r_1 = q_3 r_2 + r_3$$

...

$$r_{i-2} = q_i r_{i-1} + r_i$$

...

$$r_{k-2} = q_k r_{k-1} + r_k$$

$$r_{k-1} = q_{k+1} r_k$$

# *How to find x and y?*

x and y swapped from book, which assumes that a < b on p. 69

To find x, take

$x_0 = 1, x_1 = 0,$

$x_j = x_{j-2} - q_{j-1}x_{j-1}$

To find y, take

$y_0 = 0, y_1 = 1,$

$y_j = y_{j-2} - q_{j-1}y_{j-1}$

Use to calculate $x_k$ and $y_k$ (the desired result)

Example:

gcd(1856,5862)=2

Yields x = -101, y = 319

Check: 5862(-101) + 1856(319) = 2?

**Assume a > b**

**Let $q_i$ and $r_i$ be the series of quotients and remainders, respectively, found along the way.**

$$a = q_1 b + r_1$$

$$b = q_2 r_1 + r_2$$

$$r_1 = q_3 r_2 + r_3$$

...

$$r_{i-2} = q_i r_{i-1} + r_i$$

...

$$r_{k-2} = q_k r_{k-1} + r_k$$

$$r_{k-1} = q_{k+1} r_k$$