Class Notes 02/14/2018

Jacob Nolen

February 27, 2018

A perfect secrecy for a cryptosystem
P(the message was P)= P(the message was P | cipher was C)

Alice and Bob send the messages "Yes", "No", and "Maybe"
Frequency of messages: Yes- $\frac{5}{10}$, No- $\frac{3}{10}$, Maybe- $\frac{2}{10}$
Use a cryptosystem with 3 keys k_1, k_2, k_3

Table 1: Encryption:

	k_1	k_2	k_3
Yes	a	b	c
No	b	c	d
Maybe	c	d	a

On any given day they chose one of the keys at random. Each key is used with probability $\frac{1}{3}$. Suppose Eve captures the ciphertext "c" and wants to use this to try to decrypt the message. She wants to know if the message was "Yes".
P(message is "yes")= $\frac{1}{2}$
P(message is "yes" | cipher text is "C")

$$\frac{\text{P(message "yes" and ciphertext is "C")}}{\text{P(ciphertext is "C")}} = \frac{\frac{1}{2}\times\frac{1}{3}}{(\frac{5}{10}\times\frac{1}{3})+(\frac{3}{10}\times\frac{1}{3})+(\frac{2}{10}\times\frac{1}{3})} = \frac{1}{2}$$

So Eve learned nothing from capturing the ciphertext

Eve Captures "B"
P(message is "yes" | cipher text is "b")

$$\frac{\text{P(message "yes" and ciphertext is "C")}}{\text{P(ciphertext is "C")}} = \frac{\frac{1}{2}\times\frac{1}{3}}{(\frac{5}{10}\times\frac{1}{3})+(\frac{3}{10}\times\frac{1}{3})+0} = \frac{5}{8}$$

Since $\frac{5}{8}$ is greater than the original $\frac{1}{2}$ probability, she can give a more accurate guess. The system does not have perfect secrecy.

Theorem: The one-time-pad has perfect secrecy.

Disadvatages:

- Really long and hard to remember keys
- Can only use the key one time
- No way to transmit keys
- Impractical for actual use

Tools for elementary number theory:

How do we compute GCDs?
gcd(6,10) = 2
Factor the numbers and take all the factors they have in common.

Problem: Factoring big numbers is hard so we use Euclids Algorithm for GCDs:
If you want to find the GCD of a,b use division with remainder. Divide a by b. m is the quotient with Remainder r. This is equivalent to $a = b \times m + r$.

If r is the remainder when a is divided by b, then gcd(a,b) = gcd(b,r)

Repeat this over and over until we get a remainder of c that the last value of r is the gcd.

GCD(79,19)

$$\begin{array}{r} 4 \\ 19\overline{)79} \\ \underline{76} \\ 3 \end{array}$$

$79 = 19 \times 4 + 3$

GCD(19,3)

$$\begin{array}{r} 6 \\ 3\overline{)19} \\ \underline{18} \\ 1 \end{array}$$

$19 = 3 \times 6 + 1$

GCD(3,1)

$$\begin{array}{r} 3 \\ 1\overline{)3} \\ \underline{3} \\ 0 \end{array}$$

$3 = 1 \times 3 + 0$

SO...the GCD(79,19) = 1

Factoring method runs in the O(a + b) while Euclids algorithm runs in time O(log a + log b).

Extended Euclids Algorithm lets us compute inverses of numbers in modular arithmetic.

If GCD(a,b)=c, then there exists integers m and n such that $a \times m + b \times m = c$. We find these numbers by running Euclids Algorithm backwards.

Take each of the equations for division with remainder and solve them for the reaminder.

3 = 79 - 4(19)
1 = 1(19) - 6(3)
0 = 3 - 3(1)

Substitute 3=79-4(19)
1 = 1(19)-6(79-4(19))
1 = 1(19)-6(79) + 24(19)
1 = 25(19) - 6(79)
What is 19^{-1}(mod 79) = 25(mod 79)
reducing this equation mod79 1 = 25(19)(mod79)

Compute 7^{-1}(mod26)
Compute gcd(26,7)
26 = 3(7) + 5 ==> 5 = 26 - 3(7)
7 = 1(5) + 2 ==> 2 = 7 - 1(5)
5 = 2(2) + 1 ==> 1 = 5 - 2(2)
2 = 1(2) + 0

==> 1 = 5 - 2(2)
= 5 - 2(7 - 1(5))
= 5 - 2(7) + 2(5)
= 3(5) - 2(7)
= -2(7) + 3(26-3(7))
1 = 3(26) - 11(7)
reduce mod26
1 = -11(7)(mod26)
so... 7^{-1} = -11 = 15(mod26)

In modular arithmetic we refer to each of the possible remainders 0, 1, 2, 3...m-1 where m is our modulus as a residue (mod m)
If you have a collection of things we can add subtract and multiply (like residues) we acall this a ring. Ex:

- integers
- fractions
- real numbers
- polynomials