CSCI 2824 - CU Boulder, 2019 Summer

Lecture 17: Cryptography

1 July 2019

Lecturer: J. Marcus Hughes

Content is borrowed from Susanna Epp's Discrete Mathematics with Applications,
Rosens's Discrete Mathematics and its Applications,
Bettina and Thomas Richmond's A Discrete Transition to Advanced Mathematics, and Andrew Altomare's notes.

1 Review

- Great job completing the midterm! That was the hardest part of the class. Everything else from here is going to seem like a breeze in comparison.
- I will meet with you each individually to discuss your work and make a plan for the rest of the semester.
- You will be allowed to submit written corrections for 50% of your missed points back. You should include a description of what you did wrong originally to get full credit. Note, that you can even get points back for problems you didn't even attempt!

2 Caesar cipher

A Caesar cipher is a simple kind of crytopgraphy. We take an input of plain text and encode it by just shifting the letter. Thus, if we were shifting by 3 an A would become a D. If we assign a number to each letter, A is 1, B is 2, C is 3, and so forth we can describe this as $(M + 3) \bmod 26$. This is a very insecure scheme because you can easily brute force it. Modular arithmetic is going to be central in many cryptographic schemes.

3 More encryption

THIS SECTiON COPIED FROM ANDREW'S NOTES One of the earliest known cryptographic ciphers was used by Julius Caesar. His strategy was to shift each letter of the alphabet forward 3 places, wrapping around when you get to the end. In this scheme, for example: $A \mapsto D$, $K \mapsto N$, $Y \mapsto B$.

This is often called a **Caesar Cipher** or a **Shift Cipher**

- Mathematically, we can accomplish this by assigning to each letter a number between 0 and 25. For example:
$$A \mapsto 0, \quad K \mapsto 10, \quad Y \mapsto 24$$
- The encoding can be done by passing the value through a **shift function modulo 26**:
$$f(p) = (p + 3) \bmod 26$$

In general, for a shift k on the English alphabet we can use the function
$$f(p) = (p + k) \bmod 26$$

We can encode a message by:

1. Convert letters to numbers between 0 and 25
2. Pass each value through $f(p)$

Example: Encode $HELLO\ WORLD$ using a shift 5 cipher

1. Convert to numbers: $HELLO\ WORLD \mapsto$ 7 4 11 11 14   22 14 17 11 3
2. Shift 5: 12 9 16 16 19   1 19 22 16 8
3. Convert back to letters. The encoded message is: $MJQQT\ BTWQI$

How do we decode a message like this? If we know the shift, then it's easy–just run the message through the inverse:
$$f^{-1}(p) = (p - k) \bmod 26$$

Why is this not a very secure cipher?

The Affine Cipher

- Instead of only shifting, *multiply* and then *shift*
$$f(p) = (ap + b) \bmod 26$$
where a and b are integers with $\gcd(a, 26) = 1$
- Suppose we know a and b (i.e., we have the key)–how could we decode a message?
  - Suppose we have an encrypted character c that we know must satisfy
$$c \equiv (ap + b) \bmod 26$$

– Then we need to solve this congruence for p

$$\begin{aligned} c &\equiv (ap+b) \bmod 26 \\ \Rightarrow c-b &\equiv ap \pmod{26} \\ \Rightarrow \bar{a}(c-b) &\equiv \bar{a}ap \pmod{26} \\ &\equiv p \pmod{26} \end{aligned}$$

Example: Use an affine cipher with $a = 7$ and $b = 13$ to encrypt the letter K.

Solution: The numerical value for K is 10, so we have

$$K \mapsto a \cdot 10 + b = 7 \cdot 10 + 13 = 83 \equiv 5 \pmod{26} \mapsto F$$

Example: Find a decryption formula for this affine cipher and use it to decrypt the character F

Solution: Recall from earlier that we had the formula: $p \equiv \bar{a}(c-b) \pmod{26}$. So we need the inverse of 7 modulo 26.

$$\begin{aligned} 26 &= 3 \cdot 7 + 5 \\ 7 &= 1 \cdot 5 + 2 \\ 5 &= 2 \cdot 2 + 1 \end{aligned}$$

and in reverse

$$\begin{aligned} 1 &= 5 - 2 \cdot 2 \\ &= 5 - 2 \cdot (7 - 1 \cdot 5) = 3 \cdot 5 - 2 \cdot 7 \\ &= 3 \cdot (26 - 3 \cdot 7) - 2 \cdot 7 = 3 \cdot 26 - 11 \cdot 7 \end{aligned}$$

So the inverse of 7 modulo 26 is -11. Plugging into the decryption formula we have:

$$\begin{aligned} p &\equiv \bar{a}(c-b) \pmod{26} \\ &\equiv -11 \cdot (5 - 13) \pmod{26} \\ &\equiv 88 \pmod{26} \\ &\equiv 10 \pmod{26} \mapsto K \end{aligned}$$

So we have

$$\begin{aligned} \text{Shift Encrypt:} \quad & C = F(M) = (M+k) \bmod 26 \\ \text{Shift Decrypt:} \quad & M = F^{-1}(C) = (C-k) \bmod 26 \\ \\ \text{Affine Encrypt:} \quad & C = F(M) = (aM+b) \bmod 26 \\ \text{Affine Decrypt:} \quad & M = F^{-1}(C) = \bar{a}(C-b) \bmod 26 \end{aligned}$$

Both of these ciphers are examples of **private key encryption**. The sender and receiver both need to know the keys (k or a and b). Obviously, this poses some problems to implement on a large scale.

The solution to this: **Public Key Encryption**, which allows senders and receivers to determine secret keys by transferring public information completely in the open

RSA is the most common Public Key Encryption system. It builds on a bunch of topics we have seen before

- Fast modular exponentiation
- The Euclidean algorithm
- Bezout's theorem
- Modular inverses
- Fermat's Little theorem
- The Chinese remainder theorem

Basic idea:

- Say person X wants to send a message M to person Y
- In a *public key system*, Y will send X the public key and X will use it to encrypt the message M as cipher C
- When Y receives the message, they decrypt it using their **private key**
- Anyone in the world could intercept the public key and use it to *encrypt* a message
- But nobody could *decrypt* messages without the private key.

Implementation in practice:

- We want to create a one-way function F that, given a message M and `publicKey`, encrypts M as $C = F(M, \texttt{publicKey})$
- The function F is called "one-way" because
  - Given M and `publicKey`, it is easy to compute $C = F(m, \texttt{publicKey})$ to encrypt M
  - But if someone intercepts C and `publicKey`, it is extremely difficult to invert F and compute $M = F^{-1}(C, \texttt{publicKey})$

- Unless they have the private key, which allows the message recipient to compute $M = G(C, \texttt{privateKey})$ quickly
- So our scheme should give us functions F and G such that

  > F is easy to compute using `publicKey` but very difficult to invert, unless you know the `privateKey`, then G is easy to compute and inverts F

RSA uses a huge number n that is the product of two huge primes, p and q (~ 200 digits)

- Encryption:
  - The public key is a pair of numbers (e, n)
  - Each message M is assumed to be a number between 0 and $n - 1$
  - We encrypt a message by computing $C = M^e \bmod n$
  - which we can do quickly with fast modular exponentiation
- Decryption:
  - The private key is another pair of numbers (d, n) that relies on knowing p and q
  - We decrypt a message by computing $M = C^d \bmod n$, where d is the inverse of $e \pmod{(p-1)(q-1)}$

Example: Let's take the message $M = 7$ and encrypt it using the key $e = 11$ and $n = 35 = 7 \cdot 5$. We compute:

$$\begin{aligned} C &= M^e \bmod n \\ &= 7^{11} \bmod 35 \\ &= 7^8 \cdot 7^2 \cdot 7 \bmod 35 \\ &= (7^8 \bmod 35)(7^2 \bmod 35)(7 \bmod 35) \bmod 35 \end{aligned}$$

Lets find the binary powers of 7 modulo 35

$$\begin{aligned} 7 \bmod 35 &= 7 \\ 7^2 \bmod 35 &= 49 \bmod 35 = 14 \\ 7^4 \bmod 35 &= (7^2 \bmod 35)(7^2 \bmod 35) \bmod 35 = 21 \\ 7^8 \bmod 35 &= (7^4 \bmod 35)(7^4 \bmod 35) \bmod 35 = 21 \end{aligned}$$

So plugging these into our cipher, we have

$$C = 21 \cdot 14 \cdot 7 \bmod 35 = 2058 \bmod 35 = 28$$

So our message 7 maps to our encrypted message 28.

If an unintended recipient sees the encoded message C and the public key (e, n) it would be extremely difficult for them to find M (given that you use large primes to construct n.)

- So far we have identified the function $F: \quad F(M, (e, n)) = M^e \bmod n$. Now the idea is to find a private key (d, n) that decrypts the message via

$$C^d \bmod n = M$$

- This is our inversion function $G: \quad G(C, (d, n)) = C^d \bmod n$
- To do this, we need to find e and d that satisfy: $C^d \bmod n = (M^e)^d \bmod n = M$
- Which means we need e and d such that: $M^{ed} \bmod n = M$
- And we need to figure out how to find such a pair (e, d), where it's hard to discover d if you know (e, n)

Assumptions:

1. $n = pq$, where p and q are very large primes
2. e is a number that is relatively prime to $(p-1)(q-1)$
3. d is the **inverse** of $e \pmod{(p-1)(q-1)}$ (which must exist from #2 above)

Let's see why these assumptions are helpful

- First, notice that since $de \equiv 1 \pmod{(p-1)(q-1)}$ there exists an integer k such that

$$de = 1 + k(p-1)(q-1)$$

Now this means that for some integer k, we have that

$$C^d \equiv (M^e)^d \equiv M^{de} \equiv M^{1+k(p-1)(q-1)} \equiv M \cdot ((M^{(p-1)})^{(q-1)})^k \pmod{n}$$

In order to get our decryption function G, we need to show that this implies that

$$C^d \equiv M \pmod{n}$$

Assume that $\gcd(M, p) = 1$ and $\gcd(M, q) = 1$. Then Fermat's Little Theorem tells us that

$$M^{p-1} \equiv 1 \pmod{p} \quad \text{and} \quad M^{q-1} \equiv 1 \pmod{q}$$

Which gives (modulo p):

$$C^d \equiv M \cdot ((M^{(p-1)})^{(q-1)})^k \equiv M \cdot 1^{(q-1)k} \equiv M \pmod{p}$$

And similarly (modulo q):

$$C^d \equiv M \cdot ((M^{(q-1)})^{(p-1)})^k \equiv M \cdot 1^{(p-1)k} \equiv M \pmod{q}$$

So we have:

$$C^d \equiv M \pmod{p}$$
$$C^d \equiv M \pmod{q}$$

where $x = C^d$ is a solution to the system of congruences.

Since p and q are relatively prime (actually, both are prime), the Chinese Remainder Theorem tells us that it is a unique solution modulo $pq = n$. Thus:

$$C^d \equiv M \pmod{n}$$

Example: Encrypt and decrypt the message $M = 1819$ using a public key encryption based on $p = 43$, $q = 59$ and $e = 13$. (We have $n = 43 \cdot 59 = 2537$ and $(p-1)(q-1) = 42 \cdot 58 = 2436$)

Now we need to find the private key, d, which is the inverse of $e = 13$ modulo 2436

$$\begin{aligned} 2436 &= 13 \cdot 187 + 5 \\ 13 &= 5 \cdot 2 + 3 \\ 5 &= 3 \cdot 1 + 2 \\ 3 &= 2 \cdot 1 + 1 \end{aligned}$$

and in reverse

$$\begin{aligned} 1 &= 3 - 1 \cdot 2 \\ &= 3 - 1(5 - 1 \cdot 3) = 2 \cdot 3 - 1 \cdot 5 \\ &= 2(13 - 2 \cdot 5) - 1 \cdot 5 = 2 \cdot 13 - 5 \cdot 5 \\ &= 2 \cdot 13 - 5(2436 - 187 \cdot 13) = 937 \cdot 13 - 5 \cdot 2436 \end{aligned}$$

So, by Bezout's theorem, we find that the inverse of $e = 13$ modulo 2436 is **d=937**

We now have our public and private keys:

`publicKey` = $(e, n) = (13, 2537)$

`privateKey` = $(d, n) = (937, 2537)$

Encryption: $C = M^e \bmod n = 1819^{13} \bmod 2537 =$ (using fast mod. exp.) $= 2081$

Decryption: $M = C^d \bmod n = 2081^{937} \bmod 2537 =$ (using fast mod. exp.) $= 1819$

So why can't we break RSA?

- If we know (e, n) couldn't we do a brute-force attack to determine d?
- Technically, yes. But d is the inverse of e modulo $(p-1)(q-1)$
- Without knowing how n is factored into $n = pq$, we have no idea that we have to look for an inverse modulo $(p-1)(q-1)$
- So we'd need to factor n first
- And it turns out that factoring large products of primes is **hard**. (no known polynomial time solution)
- E.g., in 2009 researchers successfully factored a 232 decimal digit product of two primes
  - It took two years running in parallel on hundreds of machines
  - The equivalent of 2000 years running on a single core machine.

Block ciphers

When we translate our message M into a number, we need to be careful that the numerical representation of M does not exceed n

Question: Why?

Answer: If M exceeds n, then we would be unable to distinguish between M and any other message that is congruent to $M \bmod n$

Typically, we break the message up into blocks (hence, block cipher), then encode the blocks as digits, and then encrypt the blocks

Example: Suppose we want to send the message $HELP$

First, separate into blocks: $HE \quad LP$

Second, encode: $0704 \quad 1115$

- Blocks of 2 letters works for $n \geq 2525$ (since $Z = 25$)
- Blocks of 3 letters would work with $n \geq 252525$
- etc.

Lets encrypt $HELP$ using the keys from the previous example ($e = 13$, $d = 937$, $n = 2537$)

$HE = 0704^{13} \bmod 2537 \equiv 981 \pmod{2537}$

$LP = 1115^{13} \bmod 2537 \equiv 461 \pmod{2537}$

So the encrypted message would be sent out as 0981 0461

Example: Decrypt $C = 1188 \;\; 1346$ using the keys from the previous example ($e = 13$, $d = 937$, $n = 2537$)

$$1188^{937} \bmod 2537 \equiv \text{(fast mod. exp.)} \equiv 1814 \pmod{2537} \mapsto SO$$
$$1346^{937} \bmod 2537 \equiv \text{(fast mod. exp.)} \equiv 1823 \pmod{2537} \mapsto SX$$

So the decrypted message in characters would be SOSX

Convention: If your message doesn't fit perfectly into your block size, then you pad the end of the last block with X's

So this message is “SOS"

4 Equivalence relation

An equivalence relation satisfies properties of reflexivity, symmetry, and transitivitiy.

Definition: *Equivalence relation*
A relation R is an equivalence relation if it is:

- **Reflexive**: If a is in the set R relates, then aRA, i.e. a is related to itself by R.
- **Symmetry**: If a is related to b by R, then b is also related to a by R. Thus, aRb implies bRa.
- **Transitivity**: If aRb and bRc, then aRc.

For example, let R be a relation on $\mathcal{P}(S)$, where S is a finite set, defined so that R relates two sets if the have the same minimum element, then R is an equivalence relation. We see this because:

- **Reflexive**: For any set $a \subset S$ then it has a least element x. So both a and a trivially have the same minimal element so aRa.
- **Symmetry**: For any $a, b \in \mathcal{P}(S)$ if aRb then a and b have the same least element. We could also say that b and a have the same mimimal element so bRa.
- **Transitivity**: For any $a, b, c \in \mathcal{P}(S)$ if aRb and bRc then a and b have the same least element. We'll call it x. Similarly, b and c have to have the same least element, call it y. But then x and y both refer to the least element of b so $x = y$. Thus, a and c have the same least element and aRc.

5 Congruence Modulo n

What do we mean by congruence modulo n?

Definition: *Congruence modulo* n
It's a relation! Let m and n be integers and let d be a positive integer. We say that m is congruent to n modulo d and write $m \equiv n (\text{mod } d)$ if and only if $d|(m-n)$

Exercise
Prove that the following statements are equivalent for integers a, b, and n where $n > 1$:

1. $n|(a-b)$
2. $a \equiv b (\text{mod } n)$
3. $a = b + kn$ for some integer k
4. a and b have the same (nonnegative) remainder when divided by n
5. $a \mod n = b \mod n$

We can show this by showing the chain $1 \Rightarrow 2 \Rightarrow 3 \Rightarrow 4 \Rightarrow 5 \Rightarrow 1$.